Delinea

Server Suite

Documentation ⊛ 2022.x

Delinea

Table of Contents

adclient.watch.slow.lookup.info.threshold.group	667
adclient.watch.slow.lookup.info.threshold.user	668
adclient.watch.slow.lookup.warn.threshold	669
adclient.watch.slow.lookup.warn.threshold.group	670
adclient.watch.slow.lookup.warn.threshold.user	671
adclient.zone.group.count	672
addns.tcp.timeout	673
addns.wait.time	674
adjust.offset	675
audittrail.audited.command.with.args	676
audittrail.Centrify_Suite.Trusted_Path.machinecred.skipda	677
audittrail.targets	678
audittrail.	679
audittrail.	680
capi.cache.enabled	681
capi.cache.hash.table.size	682
capi.cache.log.interval	683
capi.cache.max.objects	684
capi.cache.negative.ttl	685
capi.cache.ttl	686
db2.implement.pam.ignore.users	687
db2.user.zone_enabled	688
db2.userpass.username.lower	689
dc.dead.cache.refresh	690
dc.live.cache.refresh	691
dc.penalty.time	692
dns.alive.resweep.interval	693
dns.block	694
dns.cache.negative	695
dns.cache.timeout	696
dns.dc.domain_name	697
dns.dead.resweep.interval	698
dns.gc.domain_name	699
dns.query.all.servers	700
dns.servers	701
dns.sort	702
dns.sweep.pattern	703
dns.tcp.timeout	704
dns.udp.timeouts	705
domain.dead.cache.refresh	706

This area includes the following sections:

- Install and Upgrade
- Configuration and Policy Guides
- Admin Guides
- Reports and Events
- Auditing
- User Guides
- Evaluations
- Server Suite Free
- Integrations
- Developer Tools
- Release Notes

This area includes the following sections:

- Install and Upgrade
- Configuration and Policy Guides
- Admin Guides
- Reports and Events
- Auditing
- User Guides
- Evaluations
- Server Suite Free
- Integrations
- Developer Tools
- Release Notes

Delinea

Install and Upgrade

- Planning and Deployment Guide
- Licensing Guide
- Upgrade Guide

Most large-scale deployments rely on a project team to design and articulate a project plan, and team members take on specific roles and responsibilities. Depending on your role and responsibilities, you may want to read portions of this guide selectively.

> **Note**: Most of the information in this guide applies to all platforms. However, there are some deployment scenarios and tasks that are unique to Mac OS X computers. If you manage Mac OS X computers and users, refer to the Unexpected Link Text for additional information.

The guide provides the following information:

- Planning Deployment for an Enterprise provides an overview of key concepts and the deployment lifecycle, including suggestions for who should participate in the planning process and factors to consider that will affect your deployment strategy.
- Architecture and Basic Operations describes the key components of the Server Suite software architecture and how the components work together to provide authentication and authorization services.
- Deployment Process Overview provides an overview of the steps involved in a deployment project and a preview of the tasks you can expect to complete.
- Planning organizational units and security groups discusses the Active Directory objects and organizational model that is recommended to ensure a separation of duties for UNIX administrators.
- Installing Authentication & Privilege Services provides step-by-step instructions for installing and configuring Server Suite software components on Windows computers.
- Installing Agents on Computers to be Managed describes the installation options available and provides instructions for installing Server Suite software components on UNIX and Linux computers.
- Planning to use Server Suite zones describes the importance of zones and how you can use classic and hierarchical zone for identity management, access control, and delegated administration.
- Preparing To Migrate Existing Users And Groups describes the steps to take to prepare for migrating existing users and groups, including collecting and analyzing existing profile information and creating the first zone.
- Migrating Existing Users To Hierarchical Zones describes how to import and migrate an existing user population into hierarchical zones and enable authentication using Active Directory and Server Suite software.
- Joining Computers to a Domain and Zone describes how to complete the initial migration by joining the Active Directory domain and a Server Suite zone.
- Provisioning New User and Group Profiles After Migration describes how to use the Zone Provisioning Agent and Active Directory groups to automate provisioning of new users and groups.
- Validating Operations After Deploying provides suggestions for formal testing and validation activities you can perform to move from a pilot deployment to a production environment.
- Defining Role-Based Access for Users and Computers describes the most common roles that organizations create to complete the initial deployment and how to configure the appropriate rights and assign the roles to appropriate groups.
- Migrating And Managing Service Accounts describes the strategies you can use if you want to migrate local service accounts to Active Directory to improve security for those accounts.
- Planning to Deploy in a Demilitarized Zone (DMZ) describes how to deploy Server Suite components to allow communication between a perimeter (DMZ) zone and an internal zone.
- Managing and Evolving Operations After Deployment describes management activity for operations staff and additional services you may want to implement after deployment as you evolve the Server Suite software solution.
- Templates and Sample Forms provides examples of common documents and notification messages that you can customize and use throughout the deployment process.
- Permissions Required for Administrative Tasks provides information about the specific Active Directory permissions required to perform administrative tasks on objects specific to Server Suite.

Planning Deployment for an Enterprise

This section provides a brief review of the information you should have to begin planning a successful enterprise deployment of Server Suite. It includes an overview of the deployment life cycle, roles and responsibilities to consider in assembling a deployment team, and the factors you should consider during the planning phase that will affect how you deploy Centrify software.

For an overview of Centrify software and an introduction to basic tasks, see the *Evaluation Guide for Linux and UNIX*. For a general introduction to identity, access, and configuration management or more detailed information about performing administrative tasks, see the *Administrator's Guide for Linux and UNIX*.

What You Should Know Before Planning a Deployment

Before you begin planning a full scale deployment of Centrify software, you should be familiar with key concepts, terminology, and components for Server Suite and Active Directory. You should also have information about your existing environment.

Before you continue planning the deployment, verify you have information about:

- How Active Directory is used to store user, group, and computer information in your organization and the Active Directory schema you currently have deployed.
- How you currently manage services and provision users for both Windows and nonWindows computers.
- How the Centrify Agent installed on a UNIX, Linux, or Mac OS X computer makes that computer part of an Active Directory domain.
- How zones enable you to manage user profiles, control access to computer and application resources, and delegate administrative tasks.

If you are not familiar with Centrify architecture and the components that make up the Server Suite, see Architecture and basic operations to be sure you understand the concepts, core components, and operations that enable Active Directory users to log on to UNIX, Linux, and Mac OS X computers. This guide assumes you also have access to the *Administrator's Guide for Linux and UNIX* and can refer to it, as needed, for additional details.

Why Planning a Deployment is Important

Because Centrify software becomes a critical part of your IT infrastructure once deployed, it is important that you plan and test your deployment strategy and validate the results you expect before placing Centrify components into a production environment.

After you deploy Centrify software in a production environment, the rights and roles you define will control whether users can log on and what they can do on specific computers if they are allowed to log on. Because preventing users from accessing critical resources or services can affect business operations, you should analyze the requirements of your environment as thoroughly as possible before moving from a pilot deployment into production.

The deployment process described in this guide is intended to help you to migrate existing users and groups to Active Directory with minimal disruption to end-user activity and ongoing business services. The recommendations presented are designed to give you flexibility and provide you with a framework for deploying that minimizes the effect of the deployment on the existing user population.

> **Note**: Planning is important regardless of whether you are deploying on Windows, UNIX, Linux, or Mac computers. However, some deployment steps can be skipped if you are only deploying on Windows computers or if you aren't migrating any local users or groups. For more information about deploying only on Windows computers, see the *Administrator's Guide for Windows*. For information that is specifically about deploying on Mac computers, see the *Administrator's Guide for Mac*.

What to Expect During Deployment

In most organizations, a deployment takes place in the following stages:

Evaluation

A primary senior analyst or small group installs the software in an isolated test environment. The main goal of this stage is to learn basic concepts, terminology, and operations and validate any specific functionality that is critical to the organization adopting the software. The lab environment also allows you to test the planned changes to system and user management processes without affecting user access. This proof-of-concept stage often takes place before the decision to purchase the software or with the decision to purchase a small number of licenses for extended testing.

Analysis and Design

During this stage, a planning team does deeper analysis into the goals and requirements of the organization, the current state of the environment, and the deployment and management options that best suit the organization. The main goal of this stage is to design how you will use zones, import user account

information, and assign rights and roles through a combination of Active Directory groups and zone definitions. Most of the information in this guide is intended to help you make those decisions and validate them in a pilot deployment.

Pilot Deployment

The pilot deployment is intended to be more robust than the evaluation stage. The pilot deployment is typically 10 to 20 computers, often with a common administrative owner or administrative group. The main goal of this stage is to verify your analysis accurately described your environment and to uncover any gaps that might have been missed or special circumstances that require adjustment to the design planned for zones, user account information, or rights and roles. You can include more than 20 computers in the pilot deployment, but limiting the number makes the initial migration of the user population more manageable while you become familiar with the process.

Testing and Validation

After deploying the software, most organizations perform at least some formal testing of specific scenarios to ensure the authentication and authorization rules they have defined operate as expected and users are not locked out of computers they need access to but are prevented from logging on where they don't have access rights. The main goal of this stage is to execute a test plan that exercises software operations in a number of different use cases.

Roll-Out Deployment

After sufficient testing and verification of the pilot deployment, the deployment team can use a software delivery method to install Centrify Agent packages on remote computers and join an Active Directory domain. Typically, the roll-out is done in phases, so that Centrify software is deployed on a set of computers in one subnet, IP range, or administrative domain, then later deployed on another set of computers on a different subnet, with a different IP range, or in a different administrative domain. The goal of this stage is to deploy in a controlled manner, so that any issues can be resolved before they affect additional users or computers.

Ongoing Management and Evolution

As your environment changes and evolves, it is likely that you will want to refine, customize, and extend your deployment and your authentication, authorization, computer, and user management policies. You may also develop or enhance scripts that automate provisioning and decommissioning of accounts, or update business processes to take advantage of additional functionality, such as integration with other tools to capture Centrify data or configuring database applications to use PAM-based authentication. The goal of this stage is continuous improvement to streamline business processes and operational efficiency.

Preparing a Deployment Team

In large organizations, the network architecture and Active Directory infrastructure is often highly complex and sophisticated. Adding UNIX, Linux, and Mac OS X computers and users to this infrastructure requires careful planning and is handled best with a clearly documented deployment plan. This guide is intended to help you develop such a plan and to suggest the issues you should consider in designing a deployment that suits your organization. For an example of what a deployment plan might look like, see Simplified environment analysis and zone design template.

Depending on the size of your organization, you might want to assemble a cross-functional deployment team to plan and implement a deployment strategy, set up and test a pilot deployment program, and refine, document, and roll-out operations across the organization. In addition, a deployment team might include project leads and IT staff members who will be responsible for maintaining and managing Server Suite and Active Directory on an ongoing basis after deployment or developers who will extend or integrate applications to work with Server Suite and Active Directory.

A typical deployment team might consist of members in the following roles:

Active Directory Enterprise or Domain Administrators

Know the structure and trust relationships of one or more Active Directory forests, including the topology of the Active Directory site and the roles of the domain controllers. These administrators may also be responsible for provisioning and decommissioning accounts or maintaining the tools for these business processes.

UNIX Administrators or Administrators with Specific Expertise

Manage access for all or specific groups of UNIX, Linux, or Mac OS X computers. These administrators may be responsible for specific resources, such as the servers that host mission-critical applications or a web farm, or have specific knowledge, such Oracle database administration or AIX administrative tools.

Security Administrators

Establish security policies and audit trails and define the procedures for securing computer resources and user account information. These administrators may also define the provisioning rules for the organization or have detailed knowledge of the existing provisioning process.

IT or Network Architects

Understand the overall layout of the organization's network, including internal connectivity and access to the Internet, firewalls, port usage, bandwidth and latency issues.

Application Developers

Write programs that require authentication and authorization services. Application developers might also include UNIX programmers who will be responsible for writing scripts to automate administrative tasks, such as creating zones or adding new users to a zone.

Functional Testers

Develop test cases for the user scenarios the deployment team wants to validate.

Centrify Administrative Operators

Use Access Manager and other consoles on Windows, UNIX command line programs, ADEdit library, or PowerShell scripts to manage users, groups, computers, or zones. These operators might be delegated administrators for specific zones after deployment or existing Active Directory administrators who add and remove users from groups or manage Active Directory containers.

Database Administrators

Install and manage database instances and control access to database records. If you are planning a deployment that includes auditing user activity, the deployment team should include at least one database administrator to plan for and create the databases that will store captured sessions and audit meta-data. A database administrator can also provide procedures and guidance for backing up, archiving, and removing historical data as appropriate for your organization's record retention policies.

Internal or External Auditors

Understand regulatory compliance requirements for the organization and industry. Auditors typically know the type of information they need and can define the reports that will satisfy their needs.

Assembling a cross-functional team with members who have expertise in working with Active Directory and Windows architecture and members who have expertise in managing UNIX, Linux, or Mac OS X servers and workstations is often a key component of a successful deployment.

Preparing Deployment Documentation

In addition to deploying the software, the deployment team should prepare materials that document the solutions they are deploying and the processes and procedures to assist others in migrating. The deployment documentation might include training materials for new users and test plans to verify a successful deployment that can be reused when updating the software after the initial deployment.

In general, members of the deployment team should focus on the following activities to prepare for a roll-out of Server Suite to a production environment:

- Document the configuration settings you plan to use and update the documentation as needed based on the pilot experience. For example, during the planning phase you might have drafted a plan for user and group filtering or access controls that in practice you find must be adjusted. The pilot deployment gives you the opportunity to implement your planned solution but change it, if needed.
- Document and prototype any deployment scripts that you intend to use and any processes or policy decisions that impact using those scripts. For example, you might want to automate the join process or how new users are added to a zone or modify existing scripts that provision users.
- Document issues that require troubleshooting during the pilot deployment and the resolution for each issue. You can collect this information as an organization-specific operations manual for IT staff.
- Prepare training materials for testers, operations personnel, and end-users based on the experience gained in the pilot deployment and tailored to your organization's specific needs and internal policies.
- Prepare test plans that sufficiently cover the types of scenarios that are specific to your organization's needs and internal policies. For example, if you plan to use group policies, your test plans should include scenarios for testing the group policies you plan to implement.
- Update planning documents, such as the zone structure or role definitions that you developed during the planning phase in response to the practical experience gained in the pilot deployment.

- Create checklists or instructions that are specific to your organization's deployment. For example, you may want to create a "site preparation checklist" that covers specific steps administrators should take before deploying, a "deployment checklist" that includes site-specific naming conventions and migration instructions, and a "handoff to operations checklist" to ensure a smooth hand-over to data center staff after deployment is complete.

Defining Goals for the Deployment

One of the first tasks of the deployment team should be to define the goals you want to achieve and the criteria you will use to measure whether you have met those goals. As part of this process, you should define:

- **The primary reason for deploying Centrify in your organization.** For example, if providing centralized directory service or a single point of account administration is your most important goal, you may make different deployment decisions than if auditing and restricting user access to specific computers is your primary goal. That is, you want to be sure the deployment addresses your most pressing concerns first.
- **Priorities for any additional goals you want to set for the deployment.** For example, you may want to transition to a rationalized namespace over time, but this may be a lower priority for your organization than moving from decentralized computer administration to delegated administration of the tasks users can perform on specific computers.
- **Any specific auditing requirements or security requirements that are unique to your organization or industry.** For example, the way you organize computers into groups may be determined by specific reports you need to produce.
- **Internal policies for how you update and distribute software.** For example, you should define how frequently you apply operating system patches and whether you automate software distribution.
- **Internal policies for how you assign UNIX attributes and Active Directory account information.** For example, you should identify how you have assigned UIDs, GIDs, and other UNIX-specific attributes and whether there are existing naming conventions for Active Directory users and groups.
- **Plans for who will manage UNIX profiles after deployment.** For example, you should identify the group or groups that will manage which UNIX users and computers and whether there will be separate UNIX and Active Directory administrators with shared responsibilities or a clearly defined division of responsibilities. In most cases, Centrify recommends a separation of duties model that enables UNIX administrators to manage zones and Active Directory administrators to manage user objects and group membership.

Architecture and Basic Operations

This section provides an overview of the Server Suite architecture and the components for Windows and non-Windows computers. It also describes the basic flow of operation when users log in or access applications, and what happens when an Active Directory domain controller goes down.

The information in this section is not required for planning a deployment. It is intended as background information that can help you understand the authentication and authorization process in some detail. If you want to proceed directly to planning the deployment, you can skip this section.

Server Suite Platform-Specific Components

Server Suite provide an integration layer between Active Directory in a Windows environment and computers running other operating systems or application environments. Because of this, Server Suite includes components for managing Active Directory-based objects in the Windows environment and agents that run on each server or workstation to be integrated into Active Directory.

Server Suite Components for Windows

On Windows, Server Suite includes management consoles and services to simplify the management and integration of Linux and UNIX computers and users into Active Directory.

The key components for Windows that you use in deployment are:

- Access Manager console
- Zone Provisioning Agent configuration panel and Windows service

There are several additional Windows components available for you to use, depending on the version of Server Suite software you install and the requirements of your environment. For example, Server Suite offers extensions for working with NIS maps and Active Directory group policies, as well as components to support a multi-tier architecture for auditing activity in user sessions and the Network Information Service to support agentless authentication service.

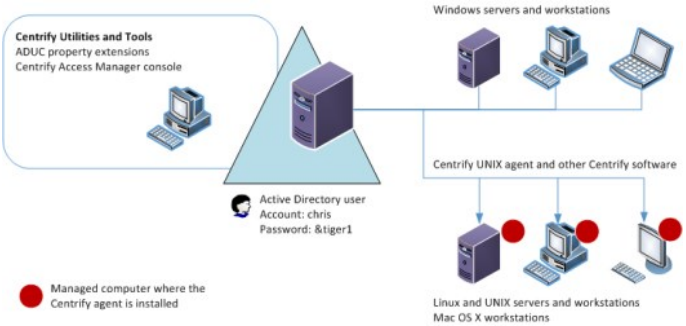
Components Installed on Managed Computers

On non-Windows computers, Server Suite software consists of the core Server Suite Agent (adclient), related libraries, and optional tools. The Server Suite Agent enables the local host computer—most commonly a Linux or UNIX computer—to join an Active Directory domain.

After the agent is deployed on a server or workstation, that computer is considered a **managed computer** and it can join any Active Directory domain you choose.

When a Server Suite-managed computer joins an Active Directory domain, it essentially becomes an Active Directory client and relies on Active Directory to provide authentication, authorization, policy management, and directory services. The interaction between the agent on the local computer and Active Directory is similar to the interaction between a Windows workstation and its Active Directory domain controller, including failover to a backup domain controller if the managed computer is unable to connect to its primary domain controller.

The following figure provides a simplified view of the integration between Windows and non-Windows computers through Server Suite software.



To use Microsoft Active Directory to centrally manage access across different platforms, you need to do the following:

- Prepare the Active Directory environment by installing the Access Manager console on at least one Windows computer and using the Setup Wizard to update the Active Directory forest.

- Ensure each UNIX, Linux, or Mac OS X computer can communicate with an appropriate Active Directory domain controller through DNS.
- Install the agent (adclient) on the UNIX, Linux, or Mac OS X computers that will be joining an Active Directory domain.
- Run the join command and specify the Active Directory domain on each UNIX, Linux, or Mac OS X computers that needs to join an Active Directory domain.
- Use Active Directory Users and Computers or Access Manager to authorize access to the UNIX, Linux, and Mac OS X computers for specific users and groups.

The next sections provide a more detailed discussion of the Server Suite architecture and a summary of what happens when a user logs on to a UNIX computer that has joined the Active Directory domain.

Storing Server Suite Properties in Active Directory

The Active Directory schema defines the object classes that can be stored in Active Directory, and the attributes that each object class must have, plus any additional attributes the object can have, and the object class that can be its parent. Schema definitions are also stored as objects in Active Directory. To store UNIX-specific attributes within the Active Directory schema, the schema must be able to include the properties that are associated with a UNIX user or group. For example, for a UNIX user, the schema needs to accommodate the following information fields:

- UNIX user name
- Password hash (optional)
- Numeric user identifier (UID)
- Primary group identifier (GID)
- General information (GECOS)
- Home directory
- Default shell

Some of these information fields are similar to standard user class attributes in Active Directory. For example, the Active Directory Display Name (displayName) attribute typically stores a user's full name—the same information typically stored in the GECOS field in an /etc/passwd file on a UNIX computer, so the displayName is used to define the contents of the GECOS field in a user's UNIX profile. Depending on the Active Directory schema you have installed, some of the information fields required for logging on to UNIX computers might not have an equivalent Active Directory attribute.

If you are using the default Active Directory schema, Server Suite stores UNIX-specific attributes in an Active Directory class under its own parent container for zones. Server Suite then organizes the information about individual UNIX computers, users, and groups by zone.

If your organization has already deployed a Microsoft-supported set of UNIX schema extensions, such as those defined in the Windows **Services for UNIX** (SFU) schema extension, you can store UNIX attributes in the fields defined by that schema as an alternative to using the zones container.

If you have deployed the **RFC 2307-compliant** Active Directory schema, you can store UNIX attributes in the fields defined by that schema and organized into RFC 2307-compliant zones.

After you have installed Server Suite components on a Windows computer, the first time you open the Access Manager administrative console, a Setup Wizard updates the Active Directory forest to include the Server Suite properties for UNIX attributes. You can then use Access Manager, the Active Directory Users and Computers MMC snap-in, ADEdit commands, or PowerShell scripts to view and modify the UNIX properties for any user, group, or computer.

> **Note**: For RFC 2307-compliant zones, the group name and UNIX name are stored in the same CN attribute. Therefore, if you change a group's name with its Active Directory Users and Computers' property page, the UNIX name is changed in Access Manager as well.

Using Access Manager

Access Manager is the primary user interface for managing all of the Server Suite-specific information stored in Active Directory. With Access Manager, you can:

- Manage access to all of your UNIX, Linux, and Mac OS X computers.
- Set and modify user and group properties for all of your UNIX, Linux, and Mac OS X users and groups.
- Create and manage zones and zone properties to simplify the process of giving users access to specific computers and migrating UNIX user accounts to Active Directory.
- Add Active Directory users and groups to zones.
- Import user and group information from local password and groups files or from NIS and NIS+ servers and domains.
- Import and maintain network information from NIS maps such as netgroup, auto.master, and automount or create custom NIS maps.
- Define and assign rights and roles that authorize or restrict access to specific computers and operations on managed computers.

You can also add other snap-ins to Access Manager or add Access Manager to another Microsoft management console snap-in. For example, you can add the

Active Directory Sites and Services and Active Directory Domains and Trusts snap-ins to Access Manager to consolidate management activity.

Allowing and Blocking Domains for Access Manager

You can configure Access Manager so that it can connect to trusted domains by setting the following registry key with a list of trusted domains and/or forests. The type of key is REG_MULTI_SZ:

HKLM\SOFTWARE\Centrify\CIMS\AllowedTrusts

Configuring a list of domains this way can be particularly useful and faster when you have a large amount of domains. Enter each domain as a separate line in the Registry Editor window.

For example, to specify a single domain:

acme.com

For example, to specify multiple domains:

acme.com
foo.com

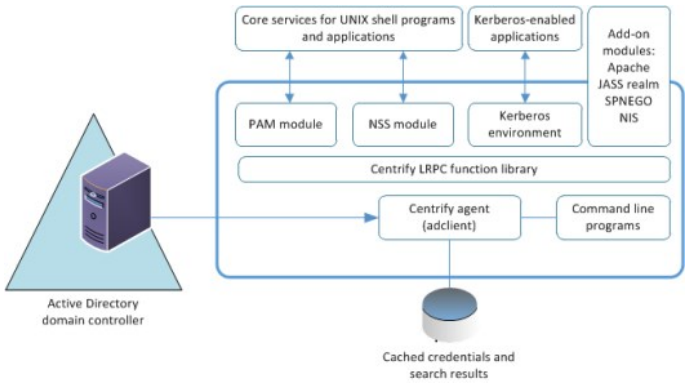
To block access to domains, you use the IgnoreTrusts key: HKLM\SOFTWARE\Centrify\CIMS\AllowedTrusts.

Core Agent Components and Services

The Server Suite Agent makes a UNIX, Linux, or Mac OS X computer look and behave like a Windows computer to Active Directory. Once installed, the agent performs the following key tasks:

- Joins UNIX, Linux, or Mac OS X computers to an Active Directory domain.
- Communicates with Active Directory to authenticate users logging on to the UNIX, Linux, or Mac OS X computer, and caches credentials for offline access.
- Enforces Active Directory authentication and password policies.
- Extends Active Directory group policies to manage the configuration of UNIX users and computers.
- Provides a Kerberos environment so that existing Kerberos applications work transparently with Active Directory.

Individual agents are platform-specific, but provide an integrated a set of services to extend Active Directory authentication, authorization, and directory service to managed computers. The following figure provides a closer look at the services provided through the Server Suite Agent:



As this figure suggests, the agent typically includes the following core components:

- The core component of the agent is the adclient process that handles all of the direct communication with Active Directory. The agent contacts Active Directory when there are requests for authentication, authorization, directory assistance, or policy updates, and then passes valid credentials or other requested information along to the programs or applications that need this information.
- The core component of the agent is the adclient process that handles all of the direct communication with Active Directory. The agent contacts Active Directory when there are requests for authentication, authorization, directory assistance, or policy updates, and then passes valid credentials or other requested information along to the programs or applications that need this information.
- The **Centrify Pluggable Authentication Module**, pam_centrifydc, enables any PAM-enabled program, such as ftpd, telnetd, login, and sshd, to

authenticate using Active Directory.

Note: For AIX and Mac OS X, the implementation is slightly different. For example, the agent for AIX can use PAM interfaces if you have configured the computer to use PAM modules or the interfaces in the Loadable Authentication Module (LAM) to handle behavior that on other platforms is done through PAM or NSS. Similarly, the agent for Mac OS X uses native interfaces where appropriate to provide services from Active Directory to the local computer.

- The **Centrify NSS** module is added to nsswitch.conf so that system look-up requests use the agent to look up and validate information using Active Directory through LDAP.
- The **ADEdit Tcl application and procedure library** and **individual UNIX command line programs** enable you to perform common administrative tasks, such as join and leave the Active Directory domain or change user passwords for Active Directory accounts interactively or within scripts to automate tasks.
- The **Server Suite-managed Kerberos environment** generates a Kerberos configuration file (etc/krb5.conf) and a default key table (krb5.keytab) file to enable your Kerberos-enabled applications to authenticate through Active Directory. These files are maintained by the agent and are updated to reflect any changes in the Active Directory forest configuration.
- The **Server Suite local cache** stores user credentials and other information for offline access and network efficiency.

In addition to these core components, the agent can also be extended with the additional software packages, including modified versions of programs such as Kerberos command line tools, OpenSSH, OpenLDAP, and PuTTY utilities. Server Suite-enabled versions of these programs allow you to use Active Directory accounts and Kerberos credentials for authentication, authorization, and policy enforcement services. Server Suite also provides authentication modules that enable you to configure single sign-on for web and database applications, and specialized extensions such as the adnisd Network Information Service, which enables you to publish information stored in Active Directory to NIS clients.

Key Operations Handled by the Adclient Process

The most important element in the agent is the adclient process. The adclient process runs as a single trusted service. This process is automatically added as a boot service and is started whenever you reboot a managed computer. The adclient process handles all of the direct communication with Active Directory and manages all of the operations provided through the other services.

The adclient process performs the following key tasks on managed computers:

- Locates the appropriate domain controllers for the local computer based on the Active Directory forest and site topology published by the Windows DNS server. If a domain controller becomes unavailable, the adclient process automatically locates the next available domain controller to ensure uninterrupted service.
- Provides Active Directory with credentials for the local computer account to verify the computer is a valid member of the domain.
- Delivers and stores user credentials so that users can be authenticated by Active Directory and, once authenticated successfully, can sign on even if the computer is disconnected from the network for mobile access or if Active Directory is unavailable.
- Caches query responses and other information, including positive and negative search results, to reduce network traffic and the number of connections to Active Directory and to ensure users can work uninterrupted and start new application sessions using their existing login credentials. All communication with Active Directory is encrypted to ensure security, and you can manage the cache through configuration parameters or group policy.
- Creates and maintains the Kerberos configuration and service ticket files to allow existing Kerberos-enabled applications to work with Active Directory without any manual configuration.
- Synchronizes the local computer's time with the clock maintained by Active Directory to ensure the timestamp on Kerberos tickets issued by the KDC are within a valid range.
- Resets the password for the local computer account in Active Directory at a regular interval to maintain security for the account's credentials.
- Provides all the authentication, authorization, and directory look-up services retrieved from Active Directory to the other Server Suite Agent services, such as the PAM service or the Apache authentication module.

How PAM Applications Work with Server Suite

Pluggable Authentication Modules (PAM) are a common mechanism for configuring authentication and authorization used by many UNIX programs and applications. If a program or application uses PAM for authentication and authorization, the rules for authenticating the user are configured in either the PAM configuration file, /etc/pam.conf or in application-specific files in the /etc/pam.d directory.

The Server Suite Agent for *NIX includes its own Pluggable Authentication Module (pam_centrifydc) that enables any application that uses PAM, such as ftpd, telnetd, login, and Apache, to authenticate users through Active Directory. When you join a domain, the pam_centrifydc module is automatically placed first in the PAM stack in systemauth, so that it takes precedence over other authentication modules.

The pam_centrifydc module is configured to work with adclient to provide a number of services, such as checking for password expiration, filtering for users and groups, and creating the local home directory and default user profile files for new users. The services provided through the pam_centrifydc module can

be customized locally on a computer, modified through Active Directory group policy, or configured through a combination of local and Active Directory settings.

Working in conjunction with the adclient process, the pam_centrifydc module provides the following services for PAM-enabled programs and applications:

- Requests the PAM-enabled application to prompt for a password when appropriate and verifies whether the application-provided user name and password are valid in Active Directory.
- Checks whether the user's password has expired in Active Directory. If the password has expired, the pam_centrifydc module prompts the user to change the password, and forwards the new password to the adclient process, which communicates the change to Active Directory.
- Queries the adclient process to determine whether any access control policies are applied. For example, the pam_centrifydc module uses the information in the centrifydc.conf file to determine whether a local user attempting to log on is mapped to an Active Directory account, whether specific users or groups have been granted or denied permission to log on to the local computer, or whether Active Directory authentication should be ignored for a specific user or group.
- Creates the local home directory and default user profile files for new users. The pam_centrifydc module uses skeleton files to set up the user environment when new Active Directory users log on to a managed computer for the first time.

Most of these tasks are performed during a user login session as a series of requests and replies from the pam_centrifydc module to Active Directory through the adclient process for those programs and applications that are configured to use PAM. Because PAM is the most common authentication service used by UNIX programs and applications, the pam_centrifydc module is the most commonly used for a typical log-on session. For a more detailed description of a typical log-on process, see What happens during the typical log-on process.

> **Note**: The order in which identity stores are listed in the nsswitch.conf file does not influence authentication. Authentication and authorization services are provided by Active Directory through the Server Suite Agent for *NIX and its PAM component, and by default, Active Directory is always tried before any other sources. The order in which sources are checked is controlled through the PAM configuration settings, for example, the lines defined in the pam.conf file. In general, you should not modify the PAM configuration because making changes to these settings can compromise security or produce unexpected and undesirable results.

How NSS Configuration Works with Server Suite

The Name Service Switch (NSS) provides a mechanism for identifying sources of network information a computer should use, such as local password and group files, NIS maps, NIS+ tables, LDAP, and DNS, and the order in which these sources should be consulted when looking up users, groups, host names, and other information.

When you join a domain, the NSS configuration file, nsswitch.conf, is automatically updated to use the Server Suite Agent's NSS module first. Using the adclient process and the service library, the Server Suite NSS module accesses network information that's stored in Active Directory through LDAP.

When a UNIX program or application needs to look up information, it checks the nsswitch.conf file and is directed to use the nss_centrifydc module. The nss_centrifydc module directs the request to Active Directory through the adclient process. The adclient process provides the information retrieved from Active Directory, then caches the information locally to ensure faster performance, reduce network traffic, and allow for disconnected operation.

> **Note**: The order in which identity stores are listed in the nsswitch.conf file does not influence authentication. Authentication and authorization services are provided by Active Directory through the Server Suite Agent and its PAM service, so Active Directory is always tried before any other sources, regardless of what you have specified in the nsswitch.conf file. Instead, the nsswitch.conf file determines the sources to use in responding to NSS queries such as getpwnam. In general, you should not modify this file because modifying the file can compromise security and complicate auditing activity. In addition, you should not specify ldap as a source in any nsswitch.conf file where you have installed the Server Suite Agent. Specifying ldap in the nsswitch.conf file can cause the system to crash.

How the Server Suite Agent Manages Kerberos Files

Kerberos is a network authentication protocol for client/server applications that uses encrypted tickets passed through a central Key Distribution Center to verify the identity of a user or service requesting access. Because Kerberos is an industry standard and a secure network authentication mechanism, you may already have UNIX programs and services that are configured to use it. To allow those existing Kerberized applications to work with Active Directory without manual configuration, the adclient process automatically creates and maintains the Kerberos configuration file, krb5.conf, and the krb5.keytab service ticket file to point Kerberos-enabled services and applications to the Key Distribution Center (KDC) in Active Directory when you join a domain.

The configuration file is initially created using information collected by probing DNS and Active Directory with the default domain set to the domain that the computer has joined. Whenever a logon or ticket validation is performed with a domain that is not in the configuration file, the configuration file is updated so that it includes the new domain. Although the adclient process can automatically update the file as needed, it does not destroy existing configuration entries that you may have added by hand. Because of this, Server Suite Agents work seamlessly with existing Kerberos-enabled applications.

> **Note**: The Authentication Service supports users defined in a Kerberos realm as long as the Kerberos domains or realms are resolvable by DNS.

Kerberos realm names are case sensitive, so be careful to check that the realm spelling and capitalization is correct. (Ref: CS-21846a)

What Happens During the Typical Log-on Process

The core Server Suite Agent for *NIX components work together to identify and authenticate the user any time a user logs on to a computer using any UNIX command that requires the user to enter credentials. The following steps summarize the interaction to help you understand the process for a typical log on request. The process is similar, though not identical, for UNIX commands that need to get information about the current user or group.

> **Note**: The following steps focus on the operation of the agent rather than the interaction between the agent and Active Directory. In addition, these steps are intended to provide a general understanding of the operations performed through the agent and do not provide a detailed analysis of a typical log on session.

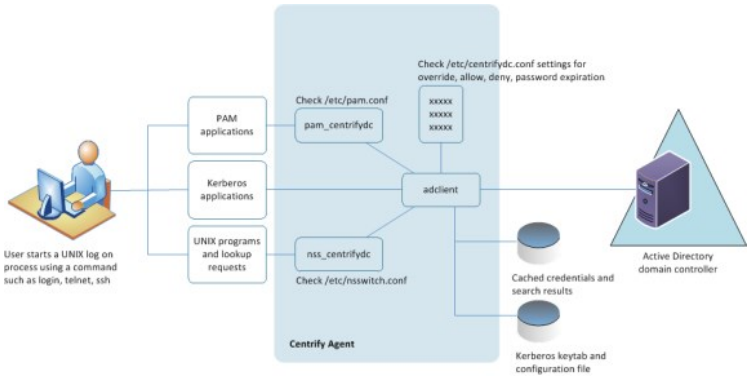
When a user starts the UNIX computer, the following takes place:

1. A login process starts and prompts the user to supply a user name.
2. The user responds by entering a valid local or Active Directory user name.
3. The login process, which is a PAM-enabled program, then reads the PAM configuration file, /etc/pam.conf, and determines that it should use the Server Suite PAM service, pam_centrifydc, for identification.
4. The login process passes the login request and the user name to the Server Suite PAM service for processing.
5. The pam_centrifydc service checks the pam.allow.override parameter in the agent configuration file to see if the user name entered is an account that should be authenticated locally.
    - If the user should be authenticated locally, the pam_centrifydc service passes the login request to the next PAM module specified in the PAM configuration file, for example, to the local configuration file /etc/passwd.
    - If the user is not listed as an override account, the pam_centrifydc service continues with the login request and checks to see if the adclient process is running, then passes the login request and user name to adclient.
6. The adclient process connects to Active Directory and queries the Active Directory domain controller to determine whether the user name included in the request is a user who has access to computers in the current computer's zone.
    - If the adclient process is unable to connect to Active Directory, it queries the local cache to determine whether the user name has been successfully authenticated before.
    - If the user account does not have access to computers in the current zone or can't be found in Active Directory or the local cache, the adclient process checks the Server Suite Agent configuration file to see if the user name is mapped to a different Active Directory user account with the adclient.mapuser.username parameter.
    - If the user name is mapped to another Active Directory account in the configuration file, the adclient process queries the Active Directory domain controller or local cache to determine whether the mapped user name has access to computers in the current computer's zone.
7. If the user has a UNIX profile for the current zone, the adclient process receives the zone-specific information for the user, such as the user's UID, the user's local UNIX name, the user's global Active Directory user name, the groups of which the user is a member, the user's home directory, and the user's default shell.
8. The adclient process checks for NSS override settings (nss.group.override and nss.user.override) to determine whether there are any changes to the user profile or additional restrictions that should override the profile retrieved or prevent the user from logging on.
9. The adclient process queries through the nss_centrifydc service to determine whether there's another user currently logged in with same UID.
    - If there is a potential conflict between local user account and the UNIX profile for an Active Directory account, the adclient process notifies the pam_centrifydc service of the potential conflict.
    - The pam_centrifydc service checks the Server Suite Agent configuration file to determine to issue a warning, ignore the conflict, or prevent the user from logging on.
    - If the login continues, the pam_centrifydc service asks the login process for a password.
10. The login process prompts the user to provide a password and returns the password entered to the pam_centrifydc service.
11. The pam_centrifydc service checks the pam.allow.users and pam.deny.users parameters in the agent configuration file to see if any user filtering has been specified to allow or deny access to specific user accounts. If any user filtering has been specified, the current user is either allowed to continue with the login process or denied access.
12. The pam_centrifydc service checks the pam.allow.groups and pam.deny.groups parameters in the agent configuration file to see if any group filtering has been specified to allow or deny access to members of specific groups. If any group filtering has been specified, the current user is either allowed to continue with the login process or denied access based on group membership.
13. If the current user account is not prevented from logging on by user or group filtering, the pam_centrifydc service queries the adclient process to see if the user is authorized to log on.
14. The adclient process queries the Active Directory domain controller through Kerberos to determine whether the user is authorized to log on to the current computer at the current time.
15. The adclient process receives the results of its authorization request from Active Directory and passes the reply to the pam_centrifydc service.
16. The pam_centrifydc service does one of the following depending on the content of the authorization reply:
    - If the user is not authorized to use the current computer or to log in at the current time, the pam_centrifydc service denies the user's request to

log on through the UNIX login process.

- If the user's password has expired, the pam_centrifydc service sends a request through the UNIX login process asking the user to change the password. After the user supplies the password, the login process completes successfully.
- If the user's password is about to expire, the pam_centrifydc service notifies the user of impending expiration through the login process.
- If the user is authorized to log on and has a current password, the login process completes successfully. If this is the first time the user has logged on to the computer through the agent, the pam_centrifydc service creates a new home directory on the computer in the location specified in the agent configuration file by the parameter pam.homeskel.dir.

The following figure provides a simplified view of a typical log-on process when using the Server Suite Agent for *NIX.



How Failover and Disconnected Access Work

The Server Suite Agent caches data from Active Directory so that users can log on and perform tasks even if the network or Active Directory server is unavailable, whether because of unexpected connectivity problems, scheduled maintenance, or offline operation of a portable computer. There are several configuration parameters that manage how the agent determines its connectivity to Active Directory, the domain controllers it should attempt to connect to, and the operation of the agent if it is unable to connect to any domain controller.

In most cases, you can set the values for the configuration parameters that control failover and disconnected operation by enabling Server Suite group policies for a site, domain, or organizational unit. Alternatively, you can set these parameters by editing the /etc/centrifydc/centrifydc.conf configuration file on individual computers.

For an overview of how the agent determines the connection status and locates a domain controller to use, see the following topics:

- Establishing a connection to DNS
- Connecting to the closest domain controller
- Restricting the domain controllers contacted
- Switching to disconnected mode
- Responding to DNS configuration changes
- Connecting to trusted forests and domains

Establishing a Connection to DNS

With each request to Active Directory, the Server Suite Agent first determines its connection status based on upon the availability of a Domain Name Service domain controller. If a DNS request for a host name takes longer than the number of seconds specified by the adclient.dns.response.maxtime parameter, the agent assumes DNS is down and switches to disconnected mode.

While running in the disconnected mode, the agent does not attempt any more synchronous network communications. Instead, it runs a background thread every 30 seconds to determine when DNS becomes available. The default value for the adclient.dns.response.maxtime is 10 seconds, but this value can be changed by group policy or by editing the /etc/centrifydc/centrifydc.conf file.

> **Note**: If the network is disconnected for a short period of time, but during that time no data is needed from Active Directory, the agent does not switch into disconnected mode. The status only changes if a connection attempt to DNS or to Active Directory through LDAP fails.

Connecting to the Closest Domain Controller

If the initial DNS request for a host name is successful, the Server Suite Agent attempts to connect to the appropriate domain controller and global catalog

for its joined domain using the **site information** found in DNS.

Site information is configured using Active Directory Sites and Services and is defined by subnet. Using the site information, the agent queries DNS for a list of the domain controllers in its site and attempts to connect to the nearest domain controller. It will continue trying to connect to each of the domain controllers in its site based on proximity until it finds a server available. If the agent is unable to connect to any of the domain controllers in its site or if no site information is available, the agent tries to connect to any remaining domain controllers listed in DNS.

Because connection status is determined by an attempt to bind to the Active Directory domain controller using an LDAP call, the adclient.ldap.socket.timeout parameter determines the maximum number of seconds the Server Suite Agent will wait for a socket connection timeout while binding to the LDAP server. The default value is 5 seconds.

Restricting the Domain Controllers Contacted

If you have a large Active Directory infrastructure or some unreliable subnets, you might want to restrict the domain controllers the agent should attempt to connect to if its primary domain controller becomes unavailable. You can limit the list of domain controllers the agent should attempt to connect to by setting the following property in the centrifydc.conf file:

dns.dc.domain_name: hostname [hostname] ...

where the domain_name is the Active Directory domain name and the hostname is a fully-qualified host name that can be resolved using DNS or the /etc/hosts file.

You can also limit the list of global catalog domain controllers the agent should attempt to connect to by setting the following property in the centrifydc.conf file:

dns.dc.forest_name: hostname [hostname] ...

where the forest_name is the forest root domain and the hostname is a fully-qualified host name that can be resolved using DNS or the /etc/hosts file.

Alternatively, you can use the adclient.server.try.max parameter or Maximum Server Connection Attempts group policy to limit the number of domain controllers the agent will attempt to connect to before switching to disconnected mode, eliminating the need to explicitly list the domain controllers using the dns.dc.domain_name and dns.gc.forest_name parameters. For example, to have the agent try a maximum of three domain controllers, you can set the following property in the centrifydc.conf file:

adclient.server.try.max: 3

Because global catalog and domain controller connections are handled independently, Server Suite Agent for *NIX can still provide authentication services if the global catalog domain controller is disconnected, as long as another domain controller is available.

Switching to Disconnected Mode

After a connection to a domain controller is established, each subsequent request for information from Active Directory checks the connection status. If a request is made to Active Directory and a response is not received within the number of seconds specified by the adclient.ldap.timeout parameter, that request is retried once. For the second request, the agent will wait up to twice as long for a response. If the second request is not answered within that amount of time, the connection to that specific domain controller is considered disconnected. Once a connection to a specific domain controller is in disconnected mode, a background thread will attempt to reconnect to that domain approximately every 30 seconds. By default, the agent waits 7 seconds for a response to the first request. If the request isn't answered, it retries the request and waits up to another 14 seconds for a response before switching to disconnected mode.

The adclient.ldap.timeout parameter specifies the maximum number of seconds to wait for Active Directory fetch, update, and delete requests to improve the response time when an initial connection attempt fails. A separate parameter, adclient.ldap.timeout.search, specifies the maximum time to wait for search requests. If the search timeout value is not specified, the default is double the adclient.ldap.timeout value. By default, therefore, the agent waits a maximum of 14 seconds for search requests.

The values for these parameters can be adjusted for high load or latency networks by configuring group policies or by editing the /etc/centrifydc/centrifydc.conf file.

Responding to DNS Configuration Changes

The DNS information collected when the agent starts and connects to a domain controller is not cached, and idle connections to Active Directory are dropped after 5 minutes by default. If you make changes in the DNS configuration, those changes are detected the next time the agent needs to reconnect, either

because an idle connection has been dropped, or the currently connected domain controller suddenly becoming unavailable.

Connecting to Trusted Forests and Domains

If the Server Suite Agent establishes a successful connection to the joined domain, it also generates or updates the /etc/krb5.conf file using the domain trust information from the global catalog, and attempts to connect to the trusted domains or to external forests to find all of the domains that are trusted.

Depending on the trust relationships you have defined, network topology, or firewall requirements, querying external trusted forests can have a significant, negative impact on network performance. You can control whether trusted domains and external forests are queried to establish transitive trusts and cross-forest authentication with the adclient.ldap.trust.enabled parameter. Setting the adclient.ldap.trust.enabled parameter to true indicates that you want the Server Suite Agent to query trusted domains and forests. Setting this parameter to false disables this feature so that the agent does not connect to any external forests or trusted domains.

If you set the adclient.ldap.trust.enabled parameter to true, you can control the maximum number of seconds to wait when searching for trust information in external forests and other trusted domains with the adclient.ldap.trust.timeout parameter. By default, the agent waits 10 seconds. The search operation is not retried if the request times out, but the request is regenerated approximately once an hour.

If your trusted domains and forests are widely distributed, have slow or unreliable network connections, or are protected by firewalls, you might want to increase the value for this parameter to allow time for the Server Suite Agent to collect information from external domains and forests.

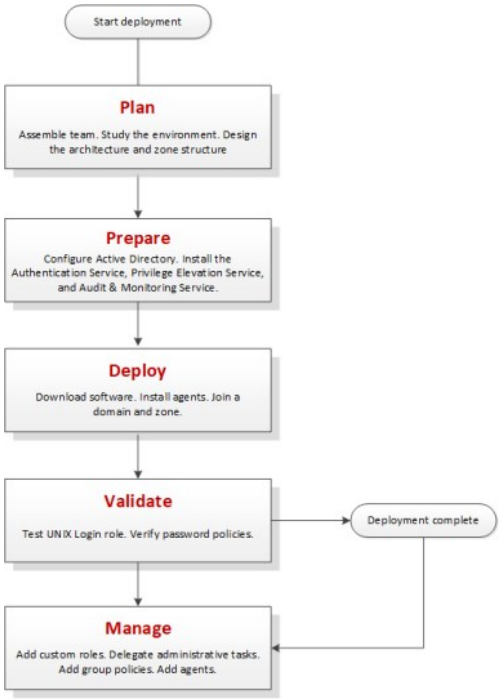
Deployment Process Overview

This chapter summarizes what's involved in deploying Centrify software. It includes simplified diagrams that highlight the steps involved and describes the tasks that are done only once, the tasks that are repeated to complete a deployment, and the tasks that may be part of the deployment project or ongoing administration after deployment.

The individual diagrams provide additional details about what's involved in each phase or the decisions you will need to make, such as who should be part of the deployment team, where to install the software, and who has permission to do what.

What's Involved in a Typical Deployment Project

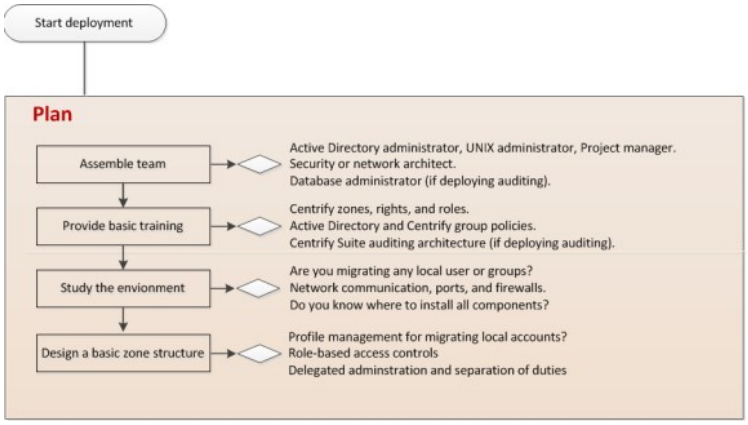
The following illustration provides a visual summary of the overall deployment process and highlights a few keys to a successful deployment.



The next sections provide additional details about each of these phases.

Plan

During the first phase of the deployment, you should collect and analyze details about your organization's requirements and goals. You can then make preliminary decisions about sizing, network communication, and what your zone structure should look like.



Here are the key steps involved:

- Assemble a deployment team with Active Directory and UNIX expertise.

  The team might also include specialists, such as database administrators, network architects, or application owners. For more information about assembling a deployment team, see Preparing a deployment team.

- Provide basic training so that members of the deployment team are familiar with Centrify concepts and terminology and know where to go for more information.

- Analyze the existing environment to determine your goals and requirements and identify target computers on which you plan to install Centrify components.

  This step is essential for designing the zone structure if you are migrating any local accounts or legacy profiles. It is also critical if you are deploying the auditing infrastructure. For more information about the questions to answer and factors that affect deployment, see Defining goals for the deployment.

- Design a basic zone structure that suits your organization.

  The zone structure depends primarily on how you want to use zones. For more information about deciding how to use zones, see Why use zones?.

- Identify a target set of computers for deployment and check that required ports are open.

Default Ports for Network Traffic and Communication

To help you plan for network traffic, the following ports are used in the initial set of network transactions when a user logs on and the agent connects to Active Directory:

- Directory Service - Global Catalog lookup request on port 3268.
- Authentication Services - LDAP sealed request on port 389.
- Kerberos – Ticket Granting Ticket (TGT) request on port 88.
- Network Time Protocol (NTP) Server – Time synchronized for Kerberos on port 123.
- Domain Name Service (DNS) – Host (A), Pointer (PTR), Service Location (SRV) records on port 53.

Depending on the specific components you deploy and operations performed, you might need to open additional ports. The following table summarizes the ports used for different editions of Centrify software.

389	Encrypted TCP/UDP communication	Centrify authentication service and privilege elevation service for Active Directory authentication and client LDAP service.
3268	Encrypted TCP communication	Centrify authentication service and privilege elevation service for Active Directory authentication and LDAP global catalog updates.

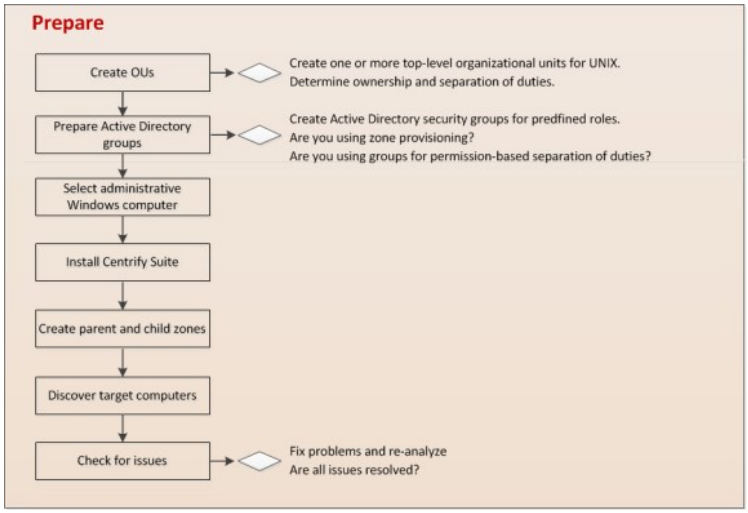
88	Encrypted UDP communication	Centrify authentication service and privilege elevation service for Kerberos ticket validation and authentication for agents and Centrify PuTTY.
464	Encrypted TCP/UDP communication for Kerberos password changes	Centrify authentication service and privilege elevation service for Kerberos ticket validation and authentication for agents, Centrify PuTTY, adpasswd, and passwd.
53	TCP/UDP communication	Centrify authentication service and privilege elevation service for clients using the Active Directory DNS server role for DNS lookup requests.
445	Encrypted TCP/UDP communication for delivery of group policies	Centrify authentication service and privilege elevation service for adclient and adgpupdate using Samba (SMB) and Windows file sharing to download and update group policies, if applicable.
123	UDP communication for simple network time protocol (NTP)	Centrify authentication service and privilege elevation service to keep time synchronized between clients and Active Directory for Kerberos ticketing.
22	Encrypted TCP communication for OpenSSH connections	Centrify authentication service and privilege elevation service to support secure shell connections on remote clients.
23	TCP communication for Telnet connections	Centrify authentication service and privilege elevation service to support telnet connections on remote clients if you cannot use secure shell (ssh). By default, telnet connections are not allowed because passwords are transferred over the network as plain text.
none	ICMP (ping) connections	Centrify authentication service and privilege elevation service to determine whether if a remote computer is reachable.
1433	Encrypted TCP communication for the collector connection to Microsoft SQL Server	Centrify authentication service, privilege elevation service, and audit and monitoring service to enable the collector service to send audited activity to the database.
5063	Encrypted TCP/RPC communication for the agent connection to collectors	Centrify authentication service, privilege elevation service, and audit and monitoring service to enable the auditing service to record user activity on an audited computer.
443	Cloud proxy server to Centrify cloud service	Centrify for mobile device management.

Network Connections and Database Management for Auditing

If you are planning a deployment with audit and monitoring service installed together with identity and privilege management, you must plan for reliable, high-speed network connections between components that collect and transfer audit data and how network traffic will be affected. You must also plan how you will create and manage the databases that store and retrieve audit data, your data archiving and retention policies, auditor permissions, and other details. For more information about planning and sizing for audit and monitoring service, see the *Auditing Administrator's Guide*.

Prepare

After you have analyzed the environment, you should prepare the Active Directory organizational units and groups to use. You can then install administrative consoles and prepare initial zones.



Here are the key steps involved:

- Create organizational units or containers to define a scope of authority.

  For example, if you want to organize all of the UNIX-related information together for your organization, you can create one top-level container for the enterprise, such as Centrify UNIX. If you want to define the scope of authority at a regional or business unit level, you might have separate top-level containers for the different regions or business units, for example, UNIX NA-SA, UNIX EMEA, UNIX PACIFIC or UNIX-Federal, UNIX-Consumer, UNIX-Industrial.

  The deployment project team should consult with the Active Directory enterprise administrator to determine the appropriate top-level containers or organizational units and who should be responsible for managing and delegating administrative tasks for the objects in those top-level containers. For more information about creating organizational units or containers in Active Directory, see Designing organizational units for Centrify.

- Create the appropriate Active Directory security groups for your organization.

  Groups can simplify permission management and the separation of duties security model. For more information about using groups, see Security groups to manage Centrify information.

- Select at least one administrative Windows computer and install Centrify components Access Manager.

  This step is not strictly required if you only use existing processes or scripts to perform administrative tasks, but Centrify recommends you have at least one computer where you can use the graphical user interface to perform common tasks. If you are deploying the audit and monitoring service infrastructure, you should also install Audit Manager and Audit Analyzer. For more information about installing Centrify software on Windows, see Installing Authentication & Privilege Services.

- Start the Centrify Access Manager console to run the Setup Wizard for the Active Directory domain.

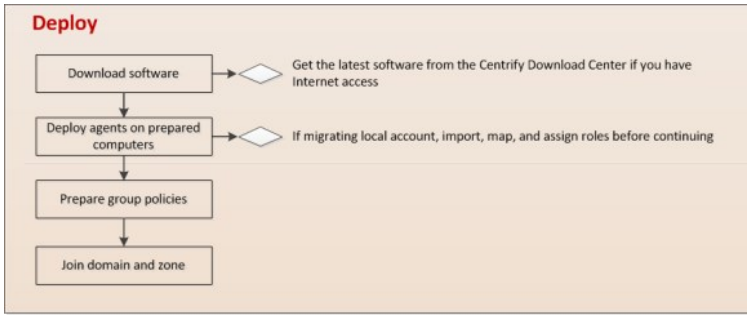
- Create a parent zone and the appropriate child zones as identified in your basic zone design.

  The hierarchical zone structure you use depends primarily on how you want to use inheritance and overrides. For more information about creating parent and child zones, see Creating the first zone.

- Determine the target set of computers and make sure that they have the appropriate connectivity.

Deploy

After you have prepared Active Directory, installed administrative consoles on at least one computer, and created at least one zone, you are ready to deploy on the computers to be managed.



Here are the key steps involved:

- Download agent software from the Centrify Download Center or a network location.
- Deploy the agent software on discovered computers that are ready for installation.
- Determine whether there are any local accounts to migrate.

  Right-click discovered computers, then click **Export Users and Groups** to generate a text file containing information about local accounts. Review the text file to determine whether there are any local accounts to migrate to Active Directory.

  If there are local accounts that must be able to log on to the discovered computer, import the groups, then users and assign them the default UNIX Login role. For more information about migrating local accounts, see Migrating existing users to hierarchical zones.
- Join the domain using the adjoin command.
- Prepare basic group policies.

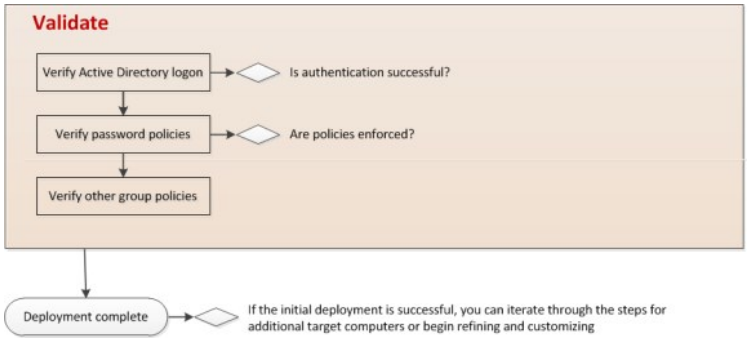
  The most common Windows computer configuration policies to deploy are:

  - Interactive Logon: Message text for users attempting to log on:—Enable and type a message that instructs the user to log on with an Active Directory user name and password.
  - Global Configuration Settings - MaxPollInterval:—Enable and set an interval if you are using Active Directory and the Centrify network time provider. Disable if you are using a native UNIX NTP daemon.
  - Enable Windows NTP Client—Enable if you are using Active Directory and the Centrify network time provider. Disable if you are using a native UNIX NTP daemon.

    The most common Centrify computer configuration policies to deploy are:
  - Set login password prompt—Enable and type a message that instructs the user to log on with an Active Directory user name and password.
  - Copy files—Enable to copy configuration files such as those required by autofs or sshd from the SYSVOL folder to managed computers.
  - Generate forwardable tickets—Disable to prevent logon tickets from being sent from one computer to another.

Validate

After you have deployed agents on target computers, you should test and verify operations before deploying on the additional computers.

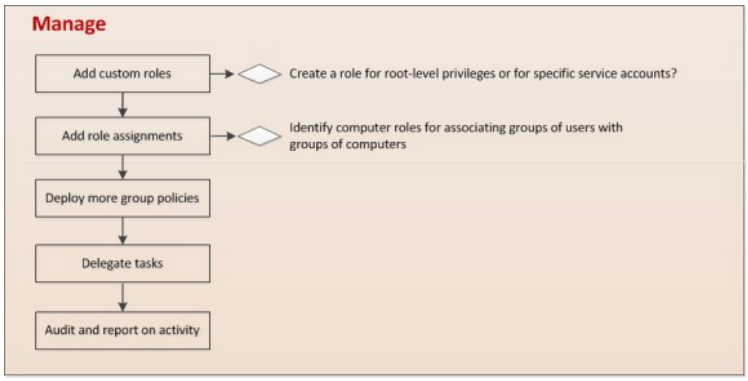


Here are the key steps involved:

- Log on to a target computer using an Active Directory user account and password to verify Active Directory authentication.
- Test password policy enforcement by attempting to change to a password that violates password complexity rules.
- Test account lockout and reset.

Manage

After you have verified the successful deployment on target computers, there are many ways you can refine, manage, and enhance on-going operations.



Here are a few of the key ways you can add value after deployment:

- Add custom roles and role assignments for users, groups, and computers.
- Import custom permissions from sudoers configuration files.
- Deploy group policies on the appropriate organizational units.
- Add the auditing infrastructure and add auditing to custom roles.
- Integrate Centrify software and Active Directory authentication and authorization services with database or web applications.

Deployment Tasks and Administrative Activity

For most deployments, there are tasks that you only perform once for an entire organization, tasks that are repeated until the deployment in complete, and tasks that are essential to deployment, but are also administrative tasks that you perform infrequently or periodically after deployment.

Steps You Only Take Once

In most organizations, you only perform the following tasks once in preparation for the deployment:

- Assemble a deployment team with Active Directory, UNIX, and other expertise.
- Provide basic training covering Centrify architecture, concepts, and terminology.
- Analyze the existing environment:
    - Find a target set of computers that share a common attribute, such as the same operating system or a similar user population.

  - Plan for permissions and the appropriate separation of duties for your organization.
  - Review network connections, ports, firewall configuration.
  - Identify computers for administration.

    Basic deployment—Access Manager

    Auditing—Audit Manager and Audit Analyzer consoles, collectors, audit databases and servers, and the installation management server

    Provisioning service—Zone Provisioning Agent and configuration tool

- Design a basic zone structure that suits your organization.
  - Single or multiple top-level parents.
  - Initial child zones, for example separate zones for Red Hat Linux and Mac OS X or different functional departments.
- Create organizational units or containers to define a scope of authority within Active Directory.
- Create Active Directory security groups for the UNIX Login role and the listed role.
- Create an Active Directory distribution group for provisioning groups and an Active Directory distribution group for provisioning users if using the provisioning service.
- Install Access Manager on at least one administrative Windows computer.
- Open Access Manager for the first time to run the Setup Wizard for the Active Directory domain.
- Create a parent zone and the appropriate child zones as identified in your basic zone design.

  Creating additional zones is an infrequent administrative task that is performed when the need arises. The basic zone design should be sufficient for the scope of your initial deployment.
- Prepare group policies to be applied.

Steps You Take More than Once During Deployment

During deployment, you perform the following tasks multiple times until you have rolled out the agent to all of the target computers that are in scope for the deployment:

- Download agent software from the Centrify Download Center or a network location.
- Deploy the agent software on computers that are ready for installation.
- If there are local accounts to migrate that must be able to log on to the discovered computer:
  - Import the groups, then users.
  - Map groups, then users to the appropriate Active Directory groups and users.
  - Assign migrated accounts the default UNIX Login role.
- Join the domain using the adjoin command.
- Verify Active Directory authentication and validate other operations.

After deployment, deploying new or updated agents is an ongoing administrative task that should be performed on a regular basis unless you have change control issues that either prevent software updates, do not allow Internet connections from the computer where Access Manager is installed, or do not want to deploy the agent on computers added to your network.

Steps You Take After Deployment to Begin Managing Zones Effectively

After you have migrated existing user populations, deployed the agent, and joined a domain, there are additional tasks you perform to complete the deployment and transition into effective zone administration.

The following tasks are optional but illustrate common administrative tasks that are often part of the deployment process to prepare for ongoing administration and improvements to operational security and efficiency:

- Create custom roles for accounts that have permission to run privileged commands.
- Create computer roles to link groups of computers with specific user role assignments.

- Map service accounts to Active Directory accounts.
- Deploy the basic set of group policies for computers and users.

What Happens After Deployment?

After deployment, ongoing management of UNIX computers, users, and groups is often handed off to Active Directory or Windows administrators or an internal service desk provisioning team to align with previously established processes and procedures for Windows servers and workstations. This is entirely a matter of organizational policy. However, in many cases UNIX administrators must continue to work with their Windows counterparts to ensure the appropriate rights and roles are assigned and the appropriate group policies are deployed.

Sample Workflow for Deployment Decisions

Centrify software solutions are extremely flexible so that they can be adapted to a wide variety of organizational requirements. All of this flexibility, however, can make deployment decisions difficult, especially in large scale or complex environments. To help you sort out the questions to ask, use the following work flow and responsibilities diagram as a guide.

This sample workflow diagram is only intended as a visual guide to the key design decisions you need to make. Many of these topics are covered in more detail in other chapters in this guide. For many organizations, however, the best guidance comes from an on-site Centrify Professional Services consultant or a Centrify partner with experience designing deployment solutions tailored to your organization's business requirements. For customized help and advice, contact your Centrify sales representative.

Planning Organizational Units and Security Groups

One of the important steps in planning a successful deployment of Server Suite is to consider how the software fits into your Active Directory infrastructure. This section describes the issues you should consider in the planning phase that affect how Active Directory and Centrify-specific objects are organized and suggests an organizational model you can use to successfully deploy Centrify within an existing Active Directory infrastructure.

If you are planning a deployment for managing and monitoring access to Windows computers, only Licenses and Zones parent containers is applicable and you can skip the other topics in this section. If you are planning a deployment that includes a mix of different platforms, however, you should review the recommendations for using organizational units (OUs) and groups.

If you plan to audit activity on any platform, Centrify recommends creating separate Active Directory security groups for auditors, administrators, and the computers that make up the audit and monitoring service infrastructure. Planning a deployment that includes the audit and monitoring service infrastructure requires additional resources and expertise. For more information about deploying auditing components, see the *Auditing Administrator's Guide*.

Identifying Stakeholders and Business Processes

Deploying Server Suite requires you to add objects to the Active Directory forest and, in most cases, update business processes for provisioning and removing users. It is important to identify who will be affected, which processes will be updated, how planned changes affect different parts of the business, and when you plan to deploy as early as possible in the planning stage.

It is also important to contact one or more Active Directory administrators to establish who will be creating the necessary objects in Active Directory and communicate the permissions required to create and manage those objects. If internal policies only allow Active Directory administrators to create organizational units (OUs) or security groups, you may need to negotiate when those activities take place and who will own the objects after they are created.

Identifying the appropriate people and processes early in the project helps eliminate unnecessary delays to the deployment and adoption of Centrify software. Communicating how the deployment affects the user and administrative communities helps ensure you can deploy rapidly and complete the project on-time.

> **Note**: The single biggest obstacle to storing UNIX data in Active Directory is overcoming internal process issues, such as change control restrictions, naming convention requirements, or proper authorization to perform administrative tasks. If you identify and resolve these challenges at the start of the project, deploying Centrify software across the enterprise becomes a fairly straightforward and painless task.

If you are a UNIX administrator, keep in mind that changes to Active Directory often require a formal change request and approval process, which can take time and delay the project. The earlier you begin planning the changes to Active Directory and the appropriate separation of duties for managing UNIX objects before, during, and after migration, the more successful the deployment will be.

Designing Organizational Units for Centrify

You can store Centrify-specific objects anywhere in the Active Directory structure if you choose. However, Centrify recommends that you create a single, high-level organizational unit (OU) specifically for Centrify objects at or near the top-level of an Active Directory forest root domain. Using one high-level organizational unit simplifies the management of Centrify containers and UNIX data.

Consolidating all UNIX data under a single organizational unit also enables you to establish an appropriate separation of duties without affecting any other previously-established OUs or permissions in Active Directory and reduces the need for additional process documentation or training. The disadvantage is that there may be strict authorization policies against setting up new organizational units.

Selecting a Location for the Top-Level OU

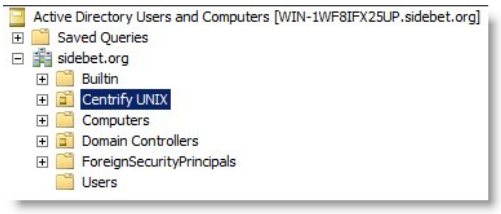
If you plan to follow the Centrify recommendation to create a top-level organizational unit for Centrify-related objects, such as Linux and UNIX computers, this high-level OU should be named so that it is easy to identify. For example, name the OU Centrify or Centrify UNIX. In deciding where this top-level OU should be placed, you should review your current Active Directory infrastructure.

There are several common scenarios:

- Single forest with a single domain
- Single forest with an empty root domain
- Single forest with account and resource domains
- Multiple forests with trust relationships
- Forests separated by a firewall (DMZ)

Single Forest with a Single Domain

If you have a single forest with a single Default-First-Site domain, you can create the top-level OU fat the same level as the default containers for Computers, Users, and other top-level objects. This is the simplest implementation of an Active Directory infrastructure. In this scenario, the top-level Centrify OU might look similar to this:



Single Forest with an Empty Root Domain

In this scenario, the forest root domain is used for the DNS namespace with one or more child domains that store information about computers, users, and groups. This is the most common Active Directory implementation. There are two important considerations if this is your Active Directory infrastructure:

- It is likely you will have a disjointed DNS namespace that you will need to resolve when you join computers to the domain.
- You might want to use multiple top-level Centrify OUs for the site. For example, assume the empty forest root domain, sidebet.org, contains three child domains:

  us.sidebet.org

  europe.sidebet.org

  asia.sidebet.org

  In this scenario, you might have a top-level Centrify OU in each child domain that includes UNIX computers to allow for more efficient data management and delegation of administrative tasks. However, if the child domains are centrally managed, you might want to create a single OU for Centrify in the forest root or one of the child domains. In general, you should base your decision on who will be responsible for managing the Centrify objects. If there are separate administrative groups for each child domain, create a top-level Centrify OU in each child domain.

Single Forest with Account and Resource Domains

In this scenario, the forest root domain has at least two child domains. One child domain stores computer principals and related information. This domain is the **resource** domain. Another child domain stores the user and group principals. This domain is the **account** domain. This scenario requires a trust relationship that allows the computers in the resource domain to trust the users and groups in the account domain. If this is your Active Directory infrastructure, you should store the Centrify data in the resource domain. For example, if the root domain, sidebet.org, contains the child domains accounts.sidebet.org and resources.sidebet.org, you would define the top-level Centrify OU in the resources.sidebet.org (OU=Acme,DC=resources,DC=sidebet,DC=org).

Multiple forests with Trust Relationships

In this scenario, you have more than one forest needing access to Centrify data. If you have multiple forests in your organizations, you should create the top-level OU for Centrify in the forests that have UNIX computers. The forests must also be configured with either a one-way or two-way trust relationship. Cross-forest authentication requires a forest functional level of Windows Server 2003 or later. Trust relationships that involve Windows NT 4.0 domains or Kerberos V5 realms are not supported.

Cross-forest Authentication for Two-way Trust Relationships

For forests that have a two-way trust relationship, users from either forest can be authenticated to log on to the other forest. For example, if you have configured a two-way trust relationship between the forest root domain sidebet.org and youbet.org, and there are both UNIX computers and UNIX users in both forests, you would create one top-level Centrify OU in each forest and users from either forest can be authenticated to the computers in either forest.

Cross-forest Authentication for One-way Trust Relationships

To allow for cross-forest authentication with a one-way trust, Centrify authenticates users in the trusted “accounts” forest to allow those users to log on to computers in the “resources” forest. Users in the trusting “resources” forest cannot log on to computers in the “accounts” forest.

For cross-forest authentication with a one-way trust, when you add the user from the "account" forest to the Centrify zone, the user's samAccountName attribute is stored in the zone object. Therefore, once the user is added to the zone, their samAccountName cannot change without causing authentication to fail.

Analyzing Trust Relationship to Prevent Authentication Failures

If your Active Directory environment does not permit one- or two-way trusts between forests, however, or uses a complex combination of one-way and two-way trust relationships between forests, users who attempt to log on from a remote forest may be denied access if the forest they are logging on to or the forest they are logging in from do not share a trust relationship.

As part of your deployment planning you should review your entire Active Directory infrastructure and determine whether you will be authenticating users from multiple forests and how trust relationships are defined for the forests users need access to. You may want to change the trust relationships you have defined.

For information about configuring trust relationships, see your Active Directory documentation.

Forests separated by a firewall (DMZ)

If you have a firewall between a forest outside of the firewall (the perimeter or DMZ forest) and a protected forest inside the firewall (the internal or corporate forest), the best security practice is to make the DMZ a separate forest with no trust relationship.

In this scenario, the top-level Centrify OU is created in the corporate forest protected by the firewall. This configuration ensures that the domain controllers in the perimeter forest cannot compromise the corporate domain controllers or get access to the corporate global catalog (GC), which stores information about all domains in the forest. Although defining no trust relationship between the perimeter forest and the corporate forest is considered the best practice for security reasons, this configuration prevents authentication through the firewall.

If you want to enable authentication for users in the corporate forest and allow them to access resources in the perimeter forest, Centrify recommends that you create a separate Active Directory forest for the computers to be placed in the network segment you are going to use as the demilitarized zone. You should then establish a one-way outgoing trust from the internal forest to the DMZ forest. For more information about deploying Centrify in a DMZ, see Planning to deploy in a demilitarized zone (DMZ).

Creating Recommended Organizational Units

In addition to the top-level Centrify OU, Centrify recommends you create several additional organizational units for managing UNIX groups, users, and computer accounts and rights and roles. These additional organizational units are intended to help you establish an appropriate separation of duties before, during, and after migration.

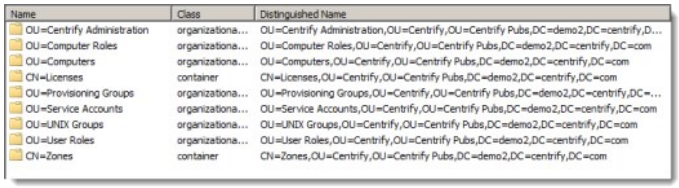
Centrify recommends the following organizational units as a starting point:

- Centrify Administration
- Computer Roles
- Computers
- Provisioning Groups
- Service Accounts
- UNIX Groups
- User Roles

Creating Organizational Units in Access Manager

If you want to create the recommended organizational unit structure for Centrify objects during your initial deployment, you can do so automatically using the Access Manager Setup Wizard. The first time you start Access Manager, it opens the Setup Wizard by default. From the Setup Wizard, you can create all of the containers for the recommended deployment structure automatically without any manual configuration. Alternatively, you can create a completely custom deployment structure by first creating a PowerShell script that creates the containers you want to use then running the custom script from within the Setup Wizard.

If you use the default script, the wizard adds the recommended organizational units and groups under the top-level **Centrify** OU and ensures all of the permissions are properly set on the objects within it. You can then select an existing organizational unit or create a new organizational unit for the components of the deployment structure. If you use the default deployment script without modification, it creates a structure like this:



Name	Class	Distinguished Name
OU=Centrify Administration	organizationa...	OU=Centrify Administration,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,D...
OU=Computer Roles	organizationa...	OU=Computer Roles,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
OU=Computers	organizationa...	OU=Computers,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
CN=Licenses	container	CN=Licenses,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
OU=Provisioning Groups	organizationa...	OU=Provisioning Groups,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=...
OU=Service Accounts	organizationa...	OU=Service Accounts,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
OU=UNIX Groups	organizationa...	OU=UNIX Groups,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
OU=User Roles	organizationa...	OU=User Roles,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com
CN=Zones	container	CN=Zones,OU=Centrify,OU=Centrify Pubs,DC=demo2,DC=centrify,DC=com

If you want to customize the script, you can use the wizard to export it. After exporting the script, you can modify it in a text editor, then restart the wizard to use the modified script. The Setup Wizard will also create the parent container for Licenses and the parent container for Zones in the Active Directory location you select.

As you roll-out the deployment, you might find additional OUs are useful. For example, you might create additional OUs because specific permissions must be granted to create, modify, delete, and manage the objects within them. By creating this organizational structure, you can control who has permission to manage the Centrify objects contained in each OU.

By using the recommended deployment structure and associated permissions, you will have a solid foundation for deploying Centrify software across the enterprise without affecting any of your existing Active Directory structure. The recommended deployment structure will also enable you to easily apply group policies and manage user, group, and computer accounts.

For more information about the purpose of the additional organizational units, see the appropriate section.

Centrify Administration Organizational Unit

The Centrify Administration OU is intended to store Active Directory security groups that ensure the separation of duties and the segregation of Centrify-related administrative operations in Active Directory.

In most cases, you will want to allow your existing Active Directory account fulfillment or provisioning team to edit UNIX groups and, potentially, the user role groups that allow for elevated permissions in UNIX. If users you have identified as Centrify Administrators are stored in the same organizational unit as the rest of the UNIX groups, then members of the fulfillment or provisioning team could grant themselves permissions to create, modify, and delete zones. With these permissions, a disgruntled provisioning staff member could delete one or more zones and prevent access to production computers. To prevent this security risk, Centrify recommends you create a separate Centrify Administration organizational unit and protect access to it.

Computer Roles Organizational Unit

The Computer Roles OU is intended to store the computer group accounts that are associated with a specific computer role. For example, if you plan to have a computer role for computers that host Oracle databases and the set of users assigned the database administrator role, you might create an Active Directory security group called Oracle_Production_Computers in this OU for the computers that host Oracle databases. If you were to add a new Oracle database instance, you would add the computer account for that database server to the Oracle_Production_Computers in this OU.

In most cases, the computer groups in this organizational unit are associated with the user role groups you add to the User Roles OU. For example, if you have a computer role for computers that host Oracle databases, you might have user role groups for database administrators and another user role group for database users. If you were to change who should be allowed to use the database or perform database administration activities, you would modify the membership of these two user role groups.

Computers Organizational Unit

The Servers OU is intended to store new computer principals in Active Directory. This organizational unit enables you to efficiently deliver computer-based group policies from Active Directory. For example, you can use a group policy to turn off SNTP updates from Active Directory for Centrify-managed computers to prevent those computers from having two registered time sources. Having a separate OU also enables you control who has the permissions and authority to create, update, and delete computer objects in the domain.

Provisioning Groups Organizational Unit

The Provisioning Groups OU is intended to store Active Directory distribution groups that are used by the Zone Provisioning Agent. The Zone Provisioning Agent is a Windows service that processes the business rules for creating or deleting UNIX profiles in zones. For example, if a new Active Directory user principal is in one of these group principals, and the group is associated with a zone, the user is automatically provisioned with a UID and a GID in that zone.

> **Note**: The profile does not allow the user to log on to computers in the zone. Identity management is separate from access management. The user's role assignments control access.

During the migration process, users you have identified as Centrify Administrators should have the appropriate permissions and authority to create, delete, and manage the membership of these Active Directory distribution groups. After migration, the team that owns the process for the provisioning UNIX accounts will need the same permissions and authority. For details about the permissions required to perform these tasks, see Setting permissions for zone groups.

Service Accounts Organizational Unit

The Service Accounts OU is intended to store the Windows service account for the Zone Provisioning Agent and any UNIX-specific service accounts that do not correlate to existing Active Directory user accounts. For example, you can use this OU for application service accounts, such as Oracle or MySQL, to enable centralized password management and auditing. By migrating service accounts from UNIX, you can centrally manage the account information, passwords, and privileges for those service accounts and their associated UNIX groups.

Unix Groups Organizational Unit

In this deployment model, the UNIX Groups OU is intended to store Active Directory security groups that are migrated from /etc/group files or NIS group maps that you want to preserve.

As part of the initial migration, you should identify one or more users as **Centrify Administrators** who should be granted the appropriate permissions and the authority to create new Active Directory group principals, and to add Active Directory user principals to the new groups.

After the migration is complete, another team might be responsible for managing the UNIX groups migrated into Active Directory. The team that owns the process for adding UNIX users to a UNIX group, removing UNIX users from a UNIX group, or creating new UNIX groups will need the permissions and the authority in Active Directory to create and delete group principals and manage the membership of those group principals in this OU (for example, ou=unix groups,ou=Centrify). For details about the permissions required to perform these tasks, see Setting permissions for zone groups.

User Roles Organizational Unit

The User Roles OU is intended to store Active Directory security groups that are associated with user role definitions that grant privileges or restrict access. For example, a user role definition might grant permission to execute commands as root or using a service account such as oracle. By associating an Active Directory group with a role definition, you can grant or deny privileges by managing Active Directory group membership.

During the migration process, users you have identified as Centrify Administrators should have the permissions and the authority to add users to the appropriate user role groups and to create new Active Directory group objects in the OU (for example, ou=user roles,ou=Centrify). After migration, your organization should decide who should be responsible for creating new user role groups and associating them with zones and who should be able to add and remove users from the User Roles organizational unit.

Licenses And Zones Parent Containers

Regardless of whether you choose to create the organizational units for the recommended deployment structure or a custom deployment structure, Centrify requires the following parent containers:

- Licenses parent container object for license keys. You must have at least one parent container for license keys in the forest. You can create more than one of these container objects to give you more granular control over who has access to which licenses.
- Zones parent container object for individual zone (ZoneName) objects. You must have at least one parent container for zones. You can create more than one parent Zones container to give you more granularity for delegating administrative tasks.

You can select the parent containers for Licenses and Zones when you run the Setup Wizard, when creating a new zone, or when managing licenses in Access Manager.

Some organizations prefer to create and manage Active Directory objects manually to ensure tight control over the objects and their attributes. For example, you might want to manually create separate parent containers for different business departments or locations if you want to manually set permissions and refine who has access to them. However, managing permissions manually can be complex and error-prone. In most cases, Centrify recommends that you establish appropriate permissions on the deployment structure and use the Zone Delegation Wizard to manage administrative permissions on individual zones and the objects contained in zones.

Security Groups To Manage Centrify Information

If you use the default recommended deployment script, the script automatically creates the following Active Directory security groups for managing Centrify-related objects:

- CentrifyAdministrators
- AuthorizationManagers
- ComputerManagers
- UnixDataManagers

Each security groups is granted the appropriate permissions to perform specific administrative tasks. For example, users who are members of the Centrify Administrators group should be able to create, modify, and delete zones.

If you are not using the recommended deployment script, you should create similar security groups for managing Centrify-related objects.

Delegating Control For Centrify Administrators

The CentrifyAdministrators security group is intended for members of the administrative or security team who are responsible for managing all Centrify-related information stored in Active Directory. You should add members to the group to grant specific users the rights required to manage Centrify licenses, zones, user roles, computer roles, provisioning groups, and the user, group, and computer profiles in each zone.

Members of the Centrify Administrators security group are responsible for identifying an organization's zone requirements and creating the zone hierarchy. Centrify Administrators also decide when to create new zones, delete obsolete zones, or consolidate existing zones. In most cases, Centrify Administrators define the basic access rights for zones and delegate administrative tasks to other users and groups on a zone-by-zone basis.

Permissions for the Centrify Administrators group are applied at the top-level of the deployment structure—for example, ou=Centrify—and grant privileges on the organizational units, containers, and object within the deployment structure. If you don't create this group or a similar security group, only members of the Domain Admins group can create new zones.

If you are managing security and individual permissions manually for Active Directory objects, see Permissions required for administrative tasks for information about the permissions required for individual tasks.

Delegating Control For Authorization Managers

The AuthorizationManagers security group is intended for members of the security team who are responsible for managing role-based access rights. You should add members to the group to grant specific users the rights required to manage user roles, computer roles, access privileges, and role assignments.

You can delegate tasks to the AuthorizationManagers group on the User Roles and Computer Roles organizational units using Active Directory Users and Computers. You can delegate zone administration tasks to the group in Access Manager.

Delegating Tasks For User Role Groups

In Active Directory Users and Computers, select the User Roles organizational unit, right-click, then select Delegate Control to start the Delegation of Control Wizard. Select the security group you are using for authorization managers and delegate the following tasks:

- Create, delete and manage groups
- Modify the membership of a group

Delegating Tasks For Computer Role Groups

In Active Directory Users and Computers, select the User Roles organizational unit, right-click, then select Delegate Control to start the Delegation of Control Wizard. Select the security group you are using for authorization managers and delegate the following tasks:

- Create, delete and manage groups
- Modify the membership of a group

Delegating Zone-specific Tasks

As a member of the CentrifyAdministrators security group, you can grant zone-specific permissions to the members of the AuthorizationManagers group. After you have created the appropriate zones, you can delegate the following zone administration tasks to authorization managers:

- Manage roles and rights
- Manage role assignments
- Modify computer roles
- Add computer roles

Delegating Control For Computer Managers

The ComputerManagers security group is intended for members of the UNIX administration team who are responsible for managing computer accounts. You should add members to this security group to grant specific users the rights required to manage computer objects in the Servers organizational unit in Active Directory.

As a member of the CentrifyAdministrators security group, you can also grant zone-specific permissions to the members of the ComputerManagers group. After you have created the appropriate zones, you can delegate the following zone administration tasks to computer managers:

- Join computers to the zone
- Remove computers from the zone

If you are managing security and individual permissions manually for Active Directory objects, see Permissions required for administrative tasks for information about the permissions required for individual tasks.

Delegating Control For Unix Data Managers

The UnixDataManagers security group is intended for members of the UNIX administration team who are responsible for managing computer accounts. You should add members to this security group to grant specific users the rights required to manage UNIX users and groups objects in the UNIX groups and Service Accounts organizational units in Active Directory.

You can delegate tasks to the UnixDataManagers group on the UNIX Groups and Service Accounts organizational units using Active Directory Users and Computers. You can delegate zone administration tasks to the group in Access Manager.

Delegating Tasks For Unix Groups

In Active Directory Users and Computers, select the UNIX Groups organizational unit, right-click, then select Delegate Control to start the Delegation of Control Wizard. Select the security group you are using for UNIX data managers and delegate the following tasks:

- Create, delete, and manage groups
- Modify the membership of a group

Delegating Tasks For Service Accounts

In Active Directory Users and Computers, select the Service Accounts organizational unit, right-click, then select Delegate Control to start the Delegation of Control Wizard. Select the security group you are using for UNIX data managers and delegate the following tasks:

- Create, delete, and manage user accounts
- Reset user passwords and force password change at next logon

Delegating Zone-Specific Tasks

As a member of the CentrifyAdministrators security group, you can also grant zone-specific permissions to the members of the UnixDataManagers group. After you have created the appropriate zones, you can delegate the following zone administration tasks to UNIX data managers:

- Add users
- Add groups
- Remove users
- Remove groups
- Modify user profiles
- Modify group profiles

Planning for Data Storage in Active Directory

Centrify stores all of the UNIX attributes required for users, groups, and computers in Active Directory, so that this information can be centrally managed. It stores these attributes without requiring you to make any modifications to the Active Directory schema you choose to use.

Changing The Zone Type

If you create a new zone using the default zone options, the new zone is created as a hierarchical zone that uses the Active Directory RFC2307-compatible schema attributes for user and group profiles. If you deselect the Use the default zone type option, you can choose to create either a hierarchical zone or a

classic zone and how you want zone information stored in the Active Directory schema.

If you are not using the default zone type and storage model, you have the following options:

- A **Standard zone** stores user and group attributes in the keywords attribute of the serviceConnectionPoint object for the user or group rather than in the user or group object.
- An **RFC2307-compatible zone** stores user and group attributes in the attributes that are defined in the RFC2307-compatible schema for user and group objects.
- An **SFU zone** stores user and group attributes in the Services for UNIX (SFU) schema attributes for the user or group object.

It is worth noting that in the default zone storage model—which uses the default Active Directory RFC2307-compatible schema—some schema attributes are not indexed. For example, in the default Active Directory RFC2307-compatible schema, the uid attribute is not an indexed attribute. Because of this limitation, queries that use this attribute might take longer than expected.

Modifying Indexed Attributes For Zones

Depending on the requirements of your organization, you might see improved performance either by creating standard Centrify zones rather than RFC2307-compatible zones or by indexing the uid attribute in default zones. Before selecting a strategy for all or selected zones, however, you should consider that indexing the uid attribute requires you to modify the Active Directory schema.

Modifying the Active Directory schema is an advanced operation that should only be performed by experienced administrators. In addition, you must be a member of the Domain Admins group or the Enterprise Admins group in Active Directory or been delegated similar authority to perform this operation. If you choose to modify the schema to improve performance, you can use "Run as ..." to select an account with appropriate permissions before performing the following steps.

To index an attribute:

1. Open a Command Prompt window, then type the following command to register the schema management assembly on your computer:

   regsvr32 schmmgmt.dll

2. Click Start, click Run, type mmc /a, then click **OK**.
3. In the console root, select the File menu, then click **Add/Remove Snap-in**.
4. Under Available Standalone Snap-ins, double-click Active Directory Schema, click **Add**, then click **OK**.
5. On the File menu, click **Save**, navigate to the %systemroot%/Sytem32 directory, type a file name for the console, then click **Save**.
6. Click **Start > All Programs** to select the Administrative Tools folder, right-click, then select **Open**.
   - If necessary, select **Organize > Layout > Menu bar** to display menus.
   - On the File menu, select **New**, then click **Shortcut**.
   - In the Create Shortcut Wizard, click in **Type the location of the item**, type the name you used for the file in Step 5, then click **Next**.
   - Select the file name in Type a name for this shortcut field, type **Active Directory Schema**, then click **Finish**.
7. Click **Start > All Programs > Administrative Tools** to select Active Directory Schema, right-click, then select **Open**.
8. Expand Attributes to select a specific attribute, such as uid, right-click, then select **Properties**.
9. Select **Index this attribute**, then click **OK**.

Viewing and Manipulating Data in Active Directory

You can view, access, and manage any information stored in Active Directory—including Centrify profiles, rights, roles, and role assignments—using ADSI Edit or using any tools that can perform standard LDAP operations such as ldifde and OpenLDAP commands such ldapsearch, ldapadd, ldapdelete and ldapmodify. For example, depending on the type of operating system and tools you prefer to use, you might view and manage Centrify profiles and zones using any combination of the following tools:

- Access Manager
- Access Module for Windows PowerShell
- Audit Module for Windows PowerShell
- Active Directory Users and Computers

- The Server Suite Windows API
- The ADEdit Tcl application and procedure library
- Centrify command-line programs

By using these tools, you can manipulate Centrify information manually or create scripts to automate key tasks such as the provisioning of new accounts. For example, you can write scripts that access the Centrify Windows API or ADEdit procedures to automatically create computer, user, or group accounts, create new zones, or assign users to roles. As part of your planning process, you should determine whether there are tasks you want to automate through the use of scripts, so that members of the development team can create or modify the appropriate tools and test them thoroughly before deploying across the organization.

Installing Authentication & Privilege Services

This section provides instructions for installing all identity and privilege management components on Windows computers in your network. There are several Windows-based components that enable you to manage the deployment and ongoing operations of Server Suite software. You should install all of the identity and privilege management components on at least one Windows computer. Depending on the division of responsibilities in your organization, you may want to install different components on more than one Windows computer.

When you install identity and privilege management components, the following features are installed:

- The Privileged Access Service, which enables MFA login, MDM, and other platform services.
- The Privilege Elevation Service and Authentication Service, which together enable computers where Server Suite software is installed to use the Active Directory infrastructure located on the domain controller, and enable users and zone-joined computers to have elevated privileges. The services include ADUC extensions, GPOE extensions, PowerShell extensions, Server Suite utilities, and Access Manager.

Access Manager is the administrative console that enables you to create zones and configure rights and roles for Active Directory users running applications on Windows computers.

You should always install the Windows components first before you install the Server Suite Agent on the non-Windows computers you intend to manage.

Preparing for Installation on Windows

Before installing Server Suite management components on Windows, you should verify that the computers where you are planning to install meet all of the system requirements and prerequisites and that you have all of the information you need to install and configure the software packages.

At a minimum, you should install the following Server Suite components on one or more Windows computers during the first stage of deployment:

- Access Manager console
- Zone Provisioning Agent

You can install these components together or independently using the setup program. Alternatively, you can install these components independently without running the setup program by using individual setup programs for each component.

Installing Server Suite

Access Manager, which is installed when you install Server Suite, is the primary management console for performing access control and privilege management operations. You typically install Access Manager directly on the computers used by one or more administrators. Alternatively, you can install it on a physical or virtual server accessed remotely by one or more administrators. The most important requirement is that the computer where you install Access Manager must be able to connect to the Active Directory domain and forest.

The Access Manager console can be installed from the setup program or from a standalone executable separate from the setup program. Before you install, you should verify your environment meets the system requirements to ensure a successful deployment.

Preparing Active Directory and DNS

All of the Server Suite software components rely on critical pieces of Active Directory infrastructure. Before you install:

- Verify Active Directory is installed and you have access to at least one Windows computer acting as a domain controller for the Active Directory forest to which you want to add UNIX computers.
- Check the configuration of DNS and whether you are using a Windows computer as the primary DNS server.
- Verify the DNS server allows secure dynamic updates and your domain controllers are configured to publish updated service locator (SRV) records.
- Verify DNS resolution and network communication between the UNIX computers and the Active Directory domain controller. You can use the ping command to test communication between the domain controller and the UNIX computer.

Identifying the Windows Computer and Log On Credentials

Depending on how you plan to manage Server Suite properties, you should identify an appropriate Windows computer and the user account credentials you should use. For example:

- Check whether the Windows computer has Active Directory Users and Computers installed.

  If you want to manage Server Suite properties using Active Directory Users and Computers, the Active Directory Users and Computers MMC snap-in

must be available on the local computer.

- Check whether the Windows computer is a domain computer, such as a Windows XP workstation, or a domain controller.

  If you install on a domain controller, you must use your own logon credential to connect to Active Directory. In most cases, you can install on any computer that has access to a domain controller.

- Verify that the Windows computer can connect to Active Directory.

- Verify that you have a Windows user account and password with sufficient rights to install software on the local computer and permission to update the Active Directory forest.

  After installation, you must be able to create new container objects in the Active Directory forest. Alternatively, an Active Directory administrator can manually configure the environment or temporarily modify your account permissions to enable you to perform setup tasks. For information about the specific rights required to perform tasks in the Setup Wizard, see Permissions required to use the Setup Wizard.

Checking Operating System and Software Requirements

Before installing on Windows, check that you have a supported version of one of the Windows operating system product families. For example, you can use Windows 7 for any console components. Alternatively, you can install components on computers in the Windows Server product family—such as Windows Server 2008 or Windows Server 2012—so that your administrative computer can be configured with additional server roles.

For more detailed information about supported platforms for specific components, see the release notes.

You should also verify that you have the .NET Framework, version 4.5 or later, installed. If the .NET Framework is not installed, the setup program can install it for you. Alternatively, you can download the .NET Framework from the Microsoft Download Center, if needed.

Checking Disk and Memory Requirements

You should also check that the computer where you are installing the Access Manager console meets the following requirements:

CPU speed	Minimum 550 MHZ
RAM	25MB
Disk space	100MB

Running the Setup Program on a Windows Computer

You can install Server Suite software using the setup program on the CD or included in the download package. The setup program copies the necessary files to the local Windows computer. There are no special permissions required to run the setup program other than permission to install files on the local computer. From the setup program, you can choose which components of you want to install.

> **Note**: If you intend to install the Zone Provisioning Agent using the setup program, you should review the requirements and other information in Installing Zone Provisioning Agent before you proceed, but you can skip the standalone installation instructions in those sections. Use the individual setup programs for components if you want to install a specific component on a specific computer. For example, use the Centrify_Zpa*version*win64.exe program to selectively install Zone Provisioning Agent components on a computer where Access Manager is not installed.

To install Authentication & Privilege on Windows:

1. Log on to the Windows computer and insert the CD or navigate to the directory where you downloaded Server Suite files.

   If the Getting Started page is not automatically displayed, double-click the autorun.exe program to start the installation of the Server Suite software.

2. On the Getting Started page, click **Authentication & Privilege** to start the setup program for identity and privilege management components.

   If any programs must be updated before installing, the setup program displays the updates required and allows you to install them. For example, you might be prompted to install or update the Microsoft .NET Framework or Microsoft SQL Server Compact edition.

3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement, click **I agree to these terms**, then click **Next**.
5. Type your name and organization, then click **Next**.
6. Expand and select the Delinea Administration and Delinea Utilities components you want to install, then click **Next**.

   If you are managing access to Linux, UNIX, and Mac OS X computers, you should select the following Server Suite Administration components for deployment:

   - **ADUC property page extensions** if you want to include Server Suite profiles when displaying properties in Active Directory Users and Computers.
   - **Access Manager** if you want to use an administrative console to manage Server Suite zones and roles.
   - **Group Policy Management Editor extension** if you want to deploy Server Suite group policies.

     You should also select the following Server Suite Utilities components for deployment:

   - **Zone Provisioning Agent** if you want to automatically provision user and group profiles into zones.

     If you want to skip the installation of any component on the local computer, click to deselect the item that you want to skip, then click **Next**. For example, if you want to skip installation of the Server Suite Reporting Service and its Microsoft SQL Server database, deselect the Server Suite Reporting Service option, then click **Next**.
7. Accept the default location for installing components, or click **Browse** to select a different location, then click **Next**.
8. Review the components you have selected, then click **Next**.

   The setup program begins installing the selected components.
9. When setup is complete for the selected packages, click **Finish** to close the setup program.

   Depending on the components you selected, you might see options to configure reporting service, the Zone Provisioning Agent, or both. You can deselect these options if you want to skip configuration or plan to install the components in a different computer. For details about configuring the Server Suite reporting service, see the *Report Administrator's Guide*. For details about configuring the Zone Provisioning Agent after installing it with the Server Suite setup program, see Configuring the Zone Provisioning Agent.

Installing Zone Provisioning Agent

The Zone Provisioning Agent enables automated provisioning of user and group accounts into Server Suite zones. You configure the Zone Provisioning Agent to monitor specific Active Directory groups that are linked to a zone. When you add or remove users or groups from the monitored groups, the Zone Provisioning Agent adds or removes corresponding users or groups in the zone.

You can install the Zone Provisioning Agent with the Server Suite setup program or as a standalone service separate from the installation of other Server Suite components. In most cases, it is installed on its own apart from the installation of other Server Suite components. After the Zone Provisioning Agent is installed, you can configure the business rules for adding and removing groups and how the attributes associated with user or group profiles are automatically generated.

About Zone Provisioning Agent and its Requirements

The Zone Provisioning Agent is intended to run on an ongoing basis on a computer that is always available. It requires a Windows user account with the right to Log on as a service. If you have a single forest, you can install the Zone Provisioning Agent on one or two computers. If you install the Zone Provisioning Agent on two computers, you should only run one instance at a time. The Zone Provisioning Agent on the second computer is intended for standby operation. You should only start the Zone Provisioning Agent on the second computer if the first instance fails.

> **Note**: The business rules that control provisioning are stored in Active Directory. If only one computer has the Zone Provisioning Agent and that computer stops running, the automated provisioning of UNIX users and group is interrupted until the computer and the Zone Provisioning Agent are restarted. Users with existing access to UNIX computers are not affected.

The Zone Provisioning Agent has the following components:

- **Zone Property Page Extension** must be installed on the same computer as the Access Manager console. This extension adds a tab to the Zone Properties for configuring provisioning rules.
- **Provisioning Agent** can be installed separately from the property page as a standalone service or on the same computer as Access Manager. The computer where you install the service should be available at all times. In most cases, this Windows service is not installed on the same computer as Access Manager.
- **Command Line Utility** can be installed separately or on the same computer as Access Manager. The command line utility allows you to write scripts for provisioning tasks or update zones on demand.

If you have more than one forest, you should install a Zone Provisioning Agent in each forest. If you have geographical domains within a single forest, you may want to install a Zone Provisioning Agent in each geographical domain. If you install a second instance of the Provisioning Agent for failover, be sure that only one instance of the Provisioning Agent runs in each forest.

Zone Provisioning Agent account permissions

Cfy_SVC_ZPA	Active Directory account	**Log on as a service**	The Zone Provisioning Agent requires permission to create UNIX profiles-- that is, the service connection points in each zone where it needs to perform provisioning operations. The service account that runs the Zone Provisioning Agent requires the **Log on as a service** right set as a local computer security policy, or in the default domain policy.

Create a service account for the Zone Provisioning Agent

The Zone Provisioning Agent must run using a valid Windows user account with the right to Log on as a service. In most cases, you should create a dedicated user account, called the **service account**, for the service to run as rather than use an existing user account.

To create a new service account for the Zone Provisioning Agent:

1. Open Active Directory Users and Computers.
2. Select the **UNIX Service Account** organizational unit.
3. Right-click, then select **New > User**.
4. Type a display name and logon name for the service account, then click **Next**.
5. Type and retype a password for the service account and modify the account options as follows, then click **Next**:
   - Uncheck **User must change password at next logon**
   - Check **User cannot change password**
   - Check **Password never expires**
6. Click **Finish** to add the service account.

Configure the local or domain group policy to allow the account to log on as a service

After you have created the service account, you must edit either a local security policy or the default domain group policy to grant the service account the **Log on as a service** right.

If you edit the default domain policy, the Zone Provisioning Agent can run on any Windows computer. If you need to move the service from one computer to another, no additional configuration is required.

Alternatively, you can edit the local security policy specifically on the computers that run the Zone Provisioning Agent. If you use the local policy, however, you may need to investigate whether other group policies are applied to the computer running the Zone Provisioning Agent to see if inheritance disables your local policy setting.

To edit the default domain group policy:

1. Open the Group Policy Object Editor and navigate to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment > Log on as a service**.
2. Right-click **Log on as a service**, then select **Properties**.
3. Select **Define these policy settings**, then click **Add User or Group**.
4. Click **Browse** to search for the service account you created.
5. Select the service account, then click **OK** to add the account and **OK** again to apply the policy.

Installing the Zone Provisioning Agent on the Access Manager computer

You can install both the Zone Provisioning Agent service and the Zone Property Page Extension on the computer where Access Manager is installed. At a minimum, you should install the Zone Property Page Extension on the same computer as the Access Manager console. The Zone Property Page Extension enables you to configure the Active Directory groups to monitor and the business rules for how to derive each user and group attribute.

> **Note**: If you select the Zone Provisioning Agent when you install components using the Server Suite setup program, all Zone Provisioning Agent components are installed. If you want to selectively install Zone Provisioning Agent components on a computer, you can install by running the Centrify_Zpaversionwin64.exe program.

To install the Zone Provisioning Agent on the Access Manager computer:

1. Log on to the Windows computer where Access Manager is installed.
2. Double-click the Centrify_Zpaversionwin64.exe file to start the Zone Provisioning Agent setup program.
3. If a User Account Control message is displayed, click **Yes**.

   If necessary, the setup program prompts you to install the Delinea Common Components.
4. On the **Welcome** screen, click **Next**.
5. Accept the licensing agreement, then click **Next**.
6. Select the features to install, then click **Next**.

   The Zone Property Page Extension is only applicable on the computer where the Access Manager console is installed. Selecting this option adds the Provisioning tab to the Zone Properties for individual zones. You can install or uncheck the other features on the computer where the Access Manager console is installed.
7. Click **Next** to accept the default location for the Zone Provisioning Agent files, or click **Browse** to select a different location, then click **Next**.
8. Click **Install** to begin installation.
9. Click **Finish** to complete the installation.

Installing the Zone Provisioning Agent on its own

You can install the Provisioning Agent as a standalone service on a computer with a relatively light load. The computer where you install the Provisioning Agent should be one that is online at all times. If the computer is shut down or suspended, the Provisioning Agent service will be suspended and no provisioning can occur. You can install a second instance of the Zone Provisioning Agent on another computer, and use your existing method of determining if a service has failed to monitor the availability of the first instance. For example, configure monitoring of the Windows Event log to notify you if the Zone Provisioning Agent service stops.

If you install the Provisioning Agent service as a standalone service, you should also install the Command Line Utility on the same computer.

To install the Zone Provisioning Agent as a standalone service:

1. Log on to the Windows computer that has a light load and is rarely shut down or offline.
2. Double-click the Centrify_Zpaversionwin64.exe file to start the Zone Provisioning Agent setup program.
3. If a User Account Control message is displayed, click **Yes**.

   If necessary, the setup program prompts you to install the Delinea Common Components.
4. On the **Welcome** screen, click **Next**.
5. Accept the licensing agreement, then click **Next**.
6. Select the Provisioning Agent and Command Line Utility features, then click **Next**.
7. Click **Next** to accept the default location for the Zone Provisioning Agent files, or click **Browse** to select a different location, then click **Next**.

8. Click **Install** to begin installation.
9. (Optional) Uncheck the **Configure and start Zone Provisioning Agent** option, then click **Finish**.

   If you leave **Configure and start Zone Provisioning Agent** selected, you are prompted to provide the service account name and password, then click **Start** to start the agent service. It is recommended that you configure the monitored containers, polling interval, and logging options, in addition to the service account name and password before starting the service. Therefore, you should open Access Manager to set up the Server Suite organization structure in Active Directory. For more information about the initial configuration, see Running Access Manager for the first time.

Configuring the Zone Provisioning Agent

By default, the Zone Provisioning Agent monitors all domains in the entire forest. If you use the recommended Server Suite organizational structure described in Creating recommended organizational units, it is recommended setting the Zone Provisioning Agent to only monitor the top-level Server Suite organizational unit or the Zones container. These objects are created in the Setup Wizard the first time you open Access Manager. After the initial configuration, you can perform the steps in this section to configure the Zone Provisioning Agent. For more information about the initial configuration, see Running Access Manager for the first time.

The most common reason for monitoring more than one organizational unit is if you have a regional or team-based OU structure in Active Directory, where each region or team is responsible for managing its own UNIX data. In this scenario, a provisioning staff member in Sidney, Australia, wouldn't be responsible for account fulfillment of a UNIX user in Chicago. To ensure the appropriate separation of duties between the different regions or teams, you would have more than one Server Suite organizational unit, and you would configure the Zone Provisioning Agent to search each of the regional organizational units.

To configure the Zone Provisioning Agent

1. Open the Zone Provisioning Agent Configuration Panel by clicking **Start > All Programs > Server Suite 2021.1 > Zone Provisioning Agent Configuration Panel**.
2. In the Monitored containers section, click **Add**.
3. Navigate to select the Server Suite organizational unit or the Zones container, then click **OK**.
4. Select **Entire Forest** *forest_name* from the list of Monitored containers, then click **Remove**.
5. Set the provisioning polling interval in minutes.

   The polling interval controls how often the Zone Provisioning Agent checks monitored containers for changes and processes the business rules for provisioning users and groups into zones. The appropriate interval often depends on the expectations of the user population or on service level agreements that define the provisioning team's commitments. In general, you should avoid polling more frequently than necessary to reduce the affect the Zone Provisioning Agent has on the performance of your domain controllers.
6. If desired, you can specify which domain controller that the Zone Provisioning Agent uses.
   1. To specify the domain controllers, click **Advanced**.

      The Advanced Domain Controller Settings dialog box displays.
   2. Click **Add** to open a separate dialog box in which you can add a domain and pick from a list of domain controllers. Click **OK** to save your chances.
   3. Click **Change** if you want to change the specified domain controller, or click **Remove** if you need to remove the specified domain controller.
7. Type the service account name or click **Browse** to locate the service account name, then type the password for the account.
8. Click **Apply**.
9. Click **Start** to start the Zone Provisioning Agent.

Whitelisting Domains for the Zone Provisioning Agent

You can configure the Zone Provisioning Agent so that it can connect to trusted domains (whitelisting) by setting the following registry key with a list of trusted domains and/or forests:

HKLM\SOFTWARE\Centrify ZPA\AllowedDomains

Configuring a list of domains this way can be particularly useful and faster when you have a large amount of domains.

For example, to specify a single domain:

HKLM\SOFTWARE\Centrify ZPA\AllowedDomains: "acme.com"

For example, to specify multiple domains:

HKLM\SOFTWARE\Centrify ZPA\AllowedDomains: "acme.com", "foo.com"

Running Access Manager for the First Time

The first time you start the Access Manager console, a Setup Wizard guides you through the initial configuration of the Active Directory forest. This initial setup creates the recommended or a custom deployment structure including the parent containers for Licenses and Zones and sets the permissions for modifying the objects within the containers. These steps are only performed once and can be done manually, if you choose.

Because the Setup Wizard creates container objects, you might need to use a domain administrator account. This requirement depends on the specific permissions your organization has configured for different classes of users. For example, if your organization only permits Domain Admins to create parent and child objects in Active Directory, you need to use an account with those permissions to run the Setup Wizard. For more information about the permissions required to perform specific configuration steps, see Permissions required to use the Setup Wizard.

Access Manager Account Permissions

n/a	Domain administrator (when running Access Manager for the first time)	domain admin (in most cases)	Because the Setup Wizard creates container objects, you might need to use a domain administrator account. This requirement depends on the specific permissions your organization has configured for different classes of users. For example, if your organization only permits Domain Admins to create parent and child objects in Active Directory, you need to use an account with those permissions to run the Setup Wizard.

To start the Setup Wizard and update the Active Directory forest

1. Open Access Manager from the desktop shortcut or Start menu.
2. Verify the name of the domain controller displayed is a member of the Active Directory forest you want to update or type the name of a different domain controller if you want to connect to a different forest, then click **OK**.
    - If you want to connect to a different forest, type the name of a domain controller in that forest.
    - If you want to connect to the forest with different credentials, select **Connect as another user**, then type a user name and password to connect as.
3. At the Welcome page, click **Next**.
4. Select **Use currently connected user credentials** to use your current log on account or select **Specify alternate user credentials** and type a user name and password, then click **Next**.
5. Select **Generate the recommended deployment structure** if you want to create all of the containers for the recommended deployment structure automatically.

    If you select this option, select whether you want to generate the default deployment structure or generate a custom structure, then click **Next**.

    - If you are generating the default structure, clicking Next enables you to select or create the location for the deployment structure in Active Directory. For example, if you want to create the top of the default deployment structure at the domain level, click **Next**, then click **Browse** to select the domain name. After you have selected a location, click **OK**. then click **Next** to create the deployment structure.
    - If you are generating a custom structure, clicking Next enables you to export the script that creates the default structure or run a script you have previously written.

      If you are generating a default or custom deployment structure, verify the successful execution of the script that creates the structure, then click **Next** to continue.

6. Verify the parent container for licenses is in the top-level Server Suite container if you are using the default deployment structure or the container of your choice, then click **Next**.

   You can add other Licenses containers in other locations later using the Manage Licenses dialog box.

   If you are not using the recommended deployment structure, the default container for license keys is domain_name/Program Data/Centrify/Licenses. To create the parent container in a different location, you can click **Browse**.

7. Review the permission requirements for the container, then click **Yes** to continue.

   If you don't want to allow the permissions for the selected container, click **No** and select a different container to continue.

8. Type or copy and paste the license key you received, then click **Add**.

   If you received multiple license keys, add each key to the list of installed licenses, then click **Next**. If you received license keys in a text file, click **Import** to import the keys directly from the file instead of adding the keys individually, then click **Next**.

   You can also add and remove license containers and keys after the initial configuration.

   For details about licensing, including how to request new license keys after deployment, check license usage and compliance, and how license counts are determined, see the *License Management Administrator's Guide*.

9. Verify that the **Create default zone container** option is selected and the parent container for zones is in the top-level Server Suite container or the container of your choice, then click **Next**.

   If you are not using the recommended deployment structure, the default container for zones is domain_name/Program Data/Centrify/Zones. To create the parent container in a different location, you can click **Browse**.

   You can skip creating the parent container in the forest or have more than one Zones parent container. For example, if you have a regional OU structure in Active Directory—where each region is responsible for its own set of zones—each region should have its own top-level organizational unit. For example, if you have separate OU structures for Tucson, AZ, and Newark, NJ, you would have separate deployment structures—SS-AZ and SS-NJ, for example—with separate parent containers for zones under each deployment structures. Users in each region can select the appropriate parent container when they create new zones.

   > **Note**: Users must have permission to read and create container objects on the parent Zones container and all child objects. You should verify the appropriate users have the permissions required to create new zones.

10. If you are using the recommended deployment structure, click **Next** to continue.

    This option allows "self-service" join operations for computers in the Computers container. It is only applicable if you are not using the recommended deployment structure. If you want to support "self-service" join operations and are not using the recommended deployment structure, select **Grant computer accounts in the Computers container permission to update their own account information**, then click **Next**.

11. If you plan to use Access Manager to manage information stored in Active Directory and maintain data integrity, click **Next** to continue.

    You should select **Register administrative notification handler for Microsoft Active Directory Users and Computers snap-in** if you want to automatically maintain the integrity of the information in Server Suite profiles.

    This option prevents Server Suite profile information from being left "orphaned" when changes are made to Active Directory objects such as users and groups. This option is not selected by default because it requires you to have Enterprise Admin or Domain Admin rights for the forest root domain.

12. Select **Activate Centrify profile property pages** if you want to be able to display Server Suite profiles in any Active Directory context, then click **Next**.

    Setting this option ensures that displaying the properties for a user, group, or computer always displays the Centrify Profile tab regardless of how you navigate to the Properties dialog box.

13. Review and confirm your configuration settings, click **Next**, then click **Finish**.

Installing Agents on Computers to be Managed

This chapter describes the recommended steps for deploying Server Suite software on the nonWindows computers that you want to add to Active Directory. The chapter also describes the alternatives you can use to install agent packages on non-Windows computers, including using native Linux installers to install Server Suite packages manually and automatically.

About the Deployment Process

The steps in this section, and in Preparing to migrate existing users and groups and Migrating existing users to hierarchical zones, are iterative in nature. In most cases, you will select a subset of computers for deployment, and repeat the steps for each target group until you have migrated all of the computers and users in the enterprise into Active Directory.

There is no technical requirement that you only work with a subset of computers at a time, but in practice the process of checking computers for potential problems and resolving open issues is more manageable when applied to a subset of computers. It is also more practical to migrate user populations in stages rather than all at once. After you step through the process a few times, you'll be able to anticipate and resolve potential issues more quickly and move into a more rapid deployment model.

Select a Target Set of Computers

As a first step in preparing to install Server Suite software, you should select a target set of computers on which to deploy. The target set can be based on any criteria you choose. In many organizations, new software must always be installed in the development environment first, then in the pre-production environment, before it can be deployed in the production environment. If your organization has this type of requirement, the first target set of computers would be the computers in the development environment.

Other possible candidates for the target set might be computers that:

- Have been identified for changes by an audit finding
- Are in the same physical location, such as a particular data center
- Share common attributes, such as all Red Hat Linux computers or all of the servers in a Web farm
- Are used by a particular department, project, or line of business
- Have a common set of users who need access to the computer resources

After you have identified a target set of computers, you are ready to begin the deployment. You should notify the user community that you are planning to install software on the target set of computers. For example, you may want to notify users by sending out an email message similar to the sample provided in Preliminary software delivery notification email template.

After you have identified a target set of computers to work with, you can use adcheck to check whether those computers have any issues that need to be resolved before you install new software on them. Checking the environment before you install helps to reduce change control issues.

Options for deploying Server Suite Agent Packages

You can:

- Run the agent installation script locally on any computer and respond to the prompts displayed.
- Create a configuration file and run the installation script remotely on any computer in silent mode.
- Use the install or update operations in the native package installer for your operating environment.
- Use a commercial or custom software distribution tool.

If you want to use one of these installation options and need more information, see the appropriate section.

Install Interactively on a Computer

The Server Suite Agent installation script, install.sh, automatically checks the operating system, disk space, DNS resolution, network connectivity, and other requirements on a target computer before installing. You can run this script interactively on any supported UNIX, Linux, or Mac OS X computer and respond to the prompts displayed.

To install Server Suite software packages on a computer interactively

1. Log on or switch to the root user if you are installing on a Linux or UNIX.

   If you are installing on Mac OS X, you can log on with any valid user account.

Note: On Mac OS X computers, you can install interactively using the graphical package installer or the install.sh script. For information about installing and joining an Active Directory domain using the Mac OS X package installer, see the Mac-specific instructions in the *Administrator's Guide for Mac*.

2. Mount the cdrom device using the appropriate command for the local computer's operating environment, if necessary. On most platforms, the CD drive is automatically mounted.

   Note: If you have downloaded the package from an FTP server or website, verify the location and go on to the next step.

   The instructions for mounting the CD drive are platform-specific. For example on Linux, you can use a command similar to the following:

   ```
   `mount /mnt/cdrom`
   ```

   To manually mount the CD drive on AIX, run a command similar to the following:

   ```
   `mount -v cdrfs -o ro /dev/cd0 /cdrom`
   ```

   To mount the CD drive on HP-UX, run a command similar to the following to display the long file names:

   ```
   mount -F cdfs -o rr /dev/dsk/c0t0d0 /mnt/cdrom
   ```

3. Change to the appropriate directory that contains the Server Suite Agent package you want to install.

   For example, to install an agent on a Linux computer from a downloaded Server Suite ISO or ZIP file, change to the Agent_Linux directory:

   ```
   cd Agent_Linux
   ```

   Similarly, if you are installing on a Solaris, HP-UX, AIX or other UNIX computer, change to the Agent_Unix directory. If installing on a Mac OS X computer, change to the Agent_Mac directory.

   If you downloaded individual agent packages from the Delinea Download Center, unzip and extract the contents. For example:

   ```
   gunzip -d centrify-infrastructure-services-VERSION-platform-arch.tgz
   ```

   ```
   tar -xf centrify-infrastructure-services-VERSION-platform-arch.tar
   ```

4. Run the install.sh script to start the installation of the agent on the local computer's operating environment. For example:

   ```
   ./install.sh
   ```

5. Follow the prompts displayed to select the services you want to install and the tasks you want to perform. For example, you can choose whether you want to:

   - Perform a default installation.
   - Perform a custom installation by selecting the specific packages to install.
   - Join a domain automatically at the conclusion of the installation.

     Depending on your selections, you may need to provide additional information, such as the user name and password for joining the domain.

Run the Bundle Installation from a Mounted Network Volume

You can install agents from a mounted network volume using the install-bundle.sh script. This script is available on the agent CD or ISO file that contains all of the supported agent platforms in compressed format. The bundle installation script automatically determines the platform required and extracts the contents of the appropriate TGZ file, then starts the normal installation process.

To use the install-bundle.sh script

1. Copy the install-bundle.sh script onto a network file system share and mount the shared directory.
2. Verify that the file is executable and that you have appropriate privileges to run it. For example:

   ```
   chmod +x install-bundle.sh
   ```

   ```
   chmod 755 install-bundle.sh
   ```

3. Run the script without command line options to start the installation or add command line options to install the agent silently.

For example, to start an interactive installation, type a command similar to this:

```
sudo ./install-bundle.sh
```

To install the agent silently, type a command similar to this:

```
./install-bundle.sh --std-suite --adjoin_opt="sidebet.org --password pa\$swd sudo ./install-bundle.sh

zone global --container sidebet.org/UNIX/Servers --server demo.sidebet.org"
```

To see complete usage information for the install-bundle.sh script, type:

```
./install-bundle.sh --help
```

Install Silently Using a Configuration File

Installing without user interaction enables you to automate software delivery and the management of remote computers. If you want to install files without any user interaction, you can run the install.sh script silently invoking the script with the appropriate command-line arguments. You can also customize the packages installed and other options by creating a custom configuration file for the installer to use.

- To see the install.sh silent mode and other command line options, enter `install.sh h`
- To install Authentication & Privilege default packages and configuration options silently, run:

  ```
  install.sh --std-suite
  ```

- To install Authentication & Privilege and Audit & Monitoring default packages and configuration options, run:

  ```
  install.sh --ent-suite
  ```

- To install a customized set of packages that all have the same version number, run:

  ```
  install.sh -n
  ```

About the Sample Configuration Files Available

You can customize the install.sh execution script. There are two sample configuration files for installing software packages silently. These sample configuration files are located in the same directory as the install.sh script:

```
centrifydc-suite.cfg

centrify-install.cfg
```

If you want to customize the packages installed or other configuration options, you can modify the sample `centrifydc-suite.cfg` or `centrifydc-install.cfg` file.

The `centrifydc-suite.cfg` file is used when you run install.sh with the `--stdsuite` or `--ent-suite` options. If you run `install.sh --std-suite` or `install.sh --ent-suite` with a customized version of the `centrifydc-suite.cfg` file, you can selectively install compatible add-on packages that do not have the same version number as the core Server Suite Agent.

Alternatively, you can run `install.sh -n` with a customized version of the `centrifydc-install.cfg` file to install the agent and add-on packages if they all have the same version number.

If you run the install.sh script silently and it cannot locate the `centrifydc-suite.cfg` or `centrifydc-install.cfg` file to use, default values defined directly in the script itself are used.

Setting the Parameters in a Custom Configuration File for the Installation Script

If you want to specify values for the install.sh script to use, you should edit the sample `centrifydc-suite.cfg` or `centrifydc-install.cfg` file in its default location before invoking the install.sh script in silent mode.

> **Note**: The parameters in the `centrifydc-install.cfg` or `centrifydc-suite.cfg` file are the same, except that the `centrifydc-suite.cfg` file is used when installing a set of services to allow packages with different version numbers to be installed together. Because you should not modify the compatibility defined in the `centrifydc-suite.cfg` file, those parameters are not included in the table.

To customize the installation using the `centrifydc-install.cfg` or `centrifydc-suite.cfg` file, you can set the following parameters:

Parameter	Description
ADCHECK	Indicate whether you want to run the `adcheck` program to check the configuration of a local computer and

its connectivity to Active Directory. Note that the install.sh script calls adcheck twice. After the first call, adcheck performs several required pre-installation steps to make sure you can install the Centrify Agent on the host computer. These steps are mandatory and cannot be skipped. However, the second call to adcheck is used to perform post-installation steps to make sure the agent has been installed successfully. The second set of checks is optional and can skipped. Set this parameter to Y if you want to run adcheck after installing. For non-interactive installations, the default is N. | | ADLICENSE | Indicate whether you want to install licensed features. Set this parameter to Y if you have purchased and installed license keys. If you downloaded and want to install unlicensed Centrify Express agents, set this parameter to N. | | GLOBAL_ZONE_ONLY | Specify whether you want to install the agent in a Solaris 10 global zone and no other zones. Set this parameter to Y only if you are running the install.sh script on a Solaris 10 computer and want to install the agent in the Solaris 10 global zone and none of your non-global zones. In most cases, you only set this parameter to Y if you use sparse root zones. The default setting for this parameter is N so that the agent is installed in all Solaris zones. If the script is not running on a Solaris 10 computer, this parameter is ignored. | | ADJOIN | Indicate whether you want to attempt to join an Active Directory domain in non-interactive mode. Set this parameter to Y to attempt to join the domain automatically. Set this parameter to N to manually join the domain after installation. | | ADJ_FORCE | Overwrite the information stored in Active Directory for an existing computer account. Set this parameter to Y to replace the information for a computer previously joined to the domain. If there is already a computer account with the same name stored in Active Directory, you must use this option if you want to replace the stored information. You should only use this option when you know it is safe to force information from the local computer to overwrite existing information. | | ADJ_TRUST | Set the **Trust for delegation** option in Active Directory for the computer account. Trusting an account for delegation allows the account to perform operations on behalf of other accounts on the network. | | DOMAIN | Specify the domain to join, if you set the ADJOIN parameter to Y. Set this parameter to the name of a valid Active Directory domain. | | USERID | Specify the Active Directory user name to use when connecting to Active Directory to join the domain. Set this parameter to a valid Active Directory user name. | | PASSWD | Specify the password for the Active Directory user name you are using to connect to Active Directory. Set this parameter to the password for the Active Directory user name specified for the USERID parameter. | | COMPUTER | Specify the computer name to use for the local host in Active Directory. Set this parameter to the computer name you want to use in Active Directory if you don't want to use the default host name for the computer. | | CONTAINER | Specify the distinguished name (DN) of the container or Organizational Unit in which you want to place this computer account. The DN you specify does not need to include the domain suffix. The domain suffix is appended programmatically to provide the complete distinguished name for the object. If you do not specify a container, the computer account is created in the domain's default Computers container. Note that the container you specify must already exist in Active Directory, and you must have permission to add entries to the specified container. | | ZONE | Specify the zone to which you want to add this computer. | | SERVER | Specify the name of the domain controller to which you prefer to connect. You can use this option to override the automatic selection of a domain controller based on the Active Directory site information. | | DA_ENABLE | Indicate whether you want to automatically enable the auditing service on the local computer. The valid settings are: Y if you want to enable auditing with the default auditing configuration. N if you don't want to enable auditing. K if you are upgrading and want to keep your current auditing configuration unchanged. | | DA_X_ENABLE | Indicate whether you want to automatically enable the Linux desktop auditing service on the local computer. The valid settings are: Y if you want to desktop enable auditing with the default auditing configuration. N if you don't want to enable desktop auditing. K if you are upgrading and want to keep your current auditing configuration unchanged | | DA_INST_NAME | Specify the name of an auditing installation if you set the DA_ENABLE parameter to Y. | | REBOOT | Indicate whether you want to automatically restart the local computer after a successful installation. Set this parameter to Y if you want to automatically restart the local computer or to N if you don't want the computer restarted automatically. | | INSTALL | Specify the operation to perform. The valid settings are: Y to install the Server Suite Agent for *NIX and any other Server Suite software packages if they are not already installed on the local computer. U to update older versions of the Server Suite Agent for *NIX and any other Server Suite packages you have installed. The update option only updates software from one major release version to another. It does not update the software if the major release version is same between packages. R to reinstall or repair the Server Suite Agent for *NIX and any other Server Suite packages you have installed. You can reinstall packages that have the same major release version but different build number or repair packages by installing an older version of the package. E to remove the software currently installed. K to keep current software unchanged. Set this parameter to Y to install or to U to update the Server Suite Agent for *NIX and other packages. If you want to install or update other packages, select the operation to perform for each package. For example to update the Server Suite Kerberos package and keep the current Server Suite LDAP proxy service, you might specify the following: CentrifyDC_krb5="U" CentrifyDC_ldapproxy="K" Note that these additional packages may have dependencies or require a specific version of the Server Suite Agent for *NIX to be installed. Before installing or updating additional packages silently, you should review the information in the *Upgrade and Compatibility Guide*. | | UNINSTALL | Specify whether you want to forcibly uninstall all installed packages. | For example, you can edit the centrifydc-install.cfg or centrifydc-suite.cfg file to silently install the Server Suite Agent for *NIX, join the domain, and automatically reboot the computer at the completion of the installation process with a file similar to this:

```
ADCHECK="N"
 ADLICENSE="Y"
 \# Solaris 10 -G option, installation in global zone only
 GLOBAL_ZONE_ONLY="N"
 ADJOIN="Y"
 ADJ_FORCE="N"
 ADJ_TRUST="N"
 DOMAIN="sample.company.com"
 USERID=administrator
 PASSWD="securepassword123"
 \#COMPUTER=my_host_name
 \#CONTAINER="my_computers"
 ZONE="global_zone"
 \#SERVER=server_name
 DA_ENABLE="N"
 DA_INST_NAME=""
 REBOOT="Y"
 \# Install the core agent package
```

```
INSTALL="Y"

\# Skip installation for other packages
CentrifyDC_nis=
CentrifyDC_krb5=
CentrifyDC_ldapproxy=
CentrifyDC_openssh=
CentrifyDC_web=
CentrifyDC_apache=
CentrifyDC_idmap=
CentrifyDA=
```

This sample configuration file does not install any of the Server Suite add-on packages. You can also use the configuration file to silently install or update selected packages. For example, to update the LDAP proxy service and OpenSSH on a computer, you would modify the configuration file to indicate that you want to update those packages:

```
CentrifyDC_ldapproxy="U"
CentrifyDC_openssh="U"
```

Customizing the Return Codes for the Installation Script

Normally, when you run the install.sh script silently, the script returns an exit code of 0 if the operation is successful. If you want the script to return exit codes that indicate whether the operation performed was a successful new installation, a successful upgrade, a successful uninstall, or there were errors preventing installation, you can also use the custom_rc option. For example:

```
install.sh -n --custom_rc
```

When you specify this option, the following return codes that are defined in the install.sh script are used to provide more detailed information about the result:

CODE_SIN=0	Successful installation
CODE_SUP=0	Successful upgrade
CODE_SUN=0	Successful uninstallation
CODE_NIN=24	Did nothing during installation
CODE_NUN=25	Did nothing during uninstallation
CODE_EIN=26	Error during installation
CODE_EUP=27	Error during upgrade
CODE_EUN=28	Error during uninstallation
CODE_ESU=29	Error encountered during setup, for example, the UID is not the root user UID, the operating environment is not supported or not recognized, or the script is executed with invalid arguments

Use Other Automated Software Distribution Utilities

You can also install Server Suite software using virtually any automated software distribution framework. For example, you can use software delivery offerings from HP OpsWare or IBM Tivoli, or features such as Apple Remote Desktop, or software distribution in the Casper Suite to deliver Server Suite software to remote computers. You can also use any custom software delivery tools you have developed specifically for your organization. If you use a commercial or custom software distribution mechanism, review the release notes text file included with agent package for platform-specific installation details.

Using a Native Package Installer

If you want to manually install a software package using a native installation program instead of the Server Suite installation script, you can follow the instructions in the *Upgrade and Compatibility Guide* for the most common native package installers, such as the Red Hat or Debian package manager. You should note that these native packages are signed with a GNU Privacy Guard (GPG) key. You need to import the key to verify the package authenticity before installing the package. You can download the RPM-GPG-KEY-centrify file from the Delinea Download Center.

Alternatively, you can use any other installation program you have available for the local operating environment. For example, if you use another program such as SMIT, YAST, APT, SUSE, or YUM to install and manage software packages, you can use that program to install Server Suite software packages.

Perform the following steps to install the Server Suite Agent using a native installation program that does not require a connection to a package repository. To use a native installation program that requires a repository connection (such as yum, SUSE or APT), see Enabling package repositories.

To install the agent using a native installation program

1. Log on as or switch to the root user.
2. If you are installing from a CD and the CD drive is not mounted automatically, use the appropriate command for the local computer's operating environment to mount the cdrom device.
3. Copy the appropriate package for the local computer's operating environment to a local directory.

   For example, if installing from the CD and the operating environment is Solaris 10 SPARC:

   ```
   cp /cdrom/cdrom0/Unix/centrifydc-release-sol10-sparc-local.tgz .
   ```

4. If the software package is a compressed file, unzip and extract the contents. For example, on Solaris:

   ```
   gunzip -d centrifydc-release-sol10-local.tgz  tar -xf centrifydc-release-sol10-sparc-local.tar
   ```

5. Run the appropriate command for installing the package based on the local computer's operating environment. For example, on Solaris:

   ```
   pkgadd -d CentrifyDC-a admin
   ```

   If you are not sure which command to use for the local operating environment, see the documentation associated with the package installer you are using.

Enabling Package Repositories

You can also download and install agents using Linux package management software for your operating system. To do this, you set up a repository for your operating system and then use the software's command line tools to manage automatic agent updates.

- **RedHat, CentOS, or Amazon systems**: Use the "Yellowdog Updater, Modified (yum)" tool to update the rpm-redhat repository. For details, see To set up and configure a RedHat, CentOS, or Amazon repository.
- **SuSE systems**: Use the Zypper tool to update the rpm-suse repository. For details, see To set up and configure a SUSE repository.
- **Debian or Ubuntu systems**: Use the Advanced Package Tool (APT), apt-get, or Aptitude tools to update the deb repository. For details, see To set up and configure a Debian or Ubuntu repository.
- **Atomic systems**: Use curl or wget to update the wget repository. For details, see To access a raw package (WGET) repository.
- **Alpine Linux systems**: For details, see To set up and configure an Alpine Linux repository.

You must perform one of the above procedures to enable the repository.

> **Note**: The procedures in this section require that you log in to the Delinea Support Portal and go to the Delinea Repo site. On that page, click the link to generate the repo key. You will then specify the repo key in a yum (RHEL, SUSE, and so forth) or APT (Debian, Ubuntu, and so forth) configuration file. There are some examples on the Delinea repo site about how to add the key to your configuration file.
>
> **Note**: For additional details about configuring and using SUSE or yum repositories, see the documentation for the distribution of Linux you are using. For additional details about configuring and using APT repositories, see the documentation for the distribution of Debian Linux or Ubuntu you are using.

WARNING: If you specify your repository on the command line, be sure to clean out your command history afterwards. Because the URL for your repository includes the credentials to access it, leaving this information around in command history is not a secure practice.

Delinea

Redhat

To Set Up and Configure a Redhat, Centos, or Amazon Repository

1. Create a /etc/yum.repos.d/centrify-rpm-redhat.repo configuration file to use the official Delinea package repository, and download a RPM-GPG-KEY-centrify key from the Delinea Support Portal.

   You can manually create the configuration file or you can use a setup script to generate the file automatically.

   Note: For the baseurl parameter, enter your Delinea repo URL token in place of **<URLtoken>**.

   - **To create the repository configuration file manually**

     Create a file with the following:

     ```
     # Source: CENTRIFY
     # Repository: CENTRIFY / rpm-redhat
     # Description: YUM repository for RedHat packages (RPMs)

     [centrify-rpm-redhat]
     name=centrify-rpm-redhat
     baseurl=https://cloudrepo.centrify.com/URLTOKEN/rpm-redhat/rpm/any-distro/any-version/$basearch
     repo_gpgcheck=1
     enabled=1
     gpgkey=https://downloads.centrify.com/products/RPM-GPG-KEY-centrify
     gpgcheck=1
     sslverify=1
     sslcacert=/etc/pki/tls/certs/ca-bundle.crt
     metadata_expire=300
     pkg_gpgcheck=1
     autorefresh=1
     type=rpm-md
     ```

- **To create the repository configuration file automatically from a script**

  ```
  curl -1sLf ' https://cloudrepo.centrify.com/URLTOKEN/rpm-redhat/cfg/setup/bash.rpm.sh' | sudo -E bash
  ```

  You should see output that lists out your repository details, such as the following example:

  ```
  # Source: CENTRIFY
  # Repository: CENTRIFY / rpm-redhat
  # Description: YUM repository for RedHat packages (RPMs)

  [centrify-rpm-redhat]
  name=centrify-rpm-redhat
  baseurl=https://cloudrepo.centrify.com/URLTOKEN/rpm-redhat/rpm/el/6/$basearch
  repo_gpgcheck=1
  enabled=1
  gpgkey=https://cloudrepo.centrify.com/URLTOKEN/rpm-redhat/cfg/gpg/gpg.BDD3FD95B65ECA48.key
  gpgcheck=1
  sslverify=1
  sslcacert=/etc/pki/tls/certs/ca-bundle.crt
  metadata_expire=300
  pkg_gpgcheck=1
  autorefresh=1
  type=rpm-md
  ```

  Note: The gpgkey listed in the output is a public key.

2. Execute the yum info command to verify the repository connection. You should see output similar to the following.

   ```
   #yum info CentrifyDC
   Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
   2020-11-24 10:20:22,669 [INFO] yum:17896:MainThread @connection.py:905 - Connection built: host=subscription.rhsm.redhat.com port=443 handler=/subscription auth=identity_cert
   ca_dir=/etc/rhsm/ca/ insecure=False
   2020-11-24 10:20:22,671 [INFO] yum:17896:MainThread @repolib.py:464 - repos updated: Repo updates

   Total repo updates: 0
   Updated
    <NONE>
   Added (new)
    <NONE>
   Deleted
    <NONE>
   This system is not registered with an entitlement server. You can use subscription-manager to register.
   Available Packages
   Name        : CentrifyDC
   ```

```
Arch        : x86_64
Version     : 5.7.0
Release     : 207
Size        : 23 M
Repo        : centrify-rpm-redhat/x86_64
Summary     : Centrify DirectControl Agent
URL         : http://www.centrify.com/
License     : BSD with portions copyright (c) Centrify Corporation 2006-2020 and licensed under Centrify End User License Agreement
Description : RPM to install Centrify DirectControl on Linux platforms.
```

3. Install the Server Suite Agent for *NIX rpm package.

```
# yum install CentrifyDC
```

> **Note**: To uninstall the Server Suite Agent rpm file, you can use the erase command. For example:

```
# yum erase CentrifyDC
```

Delinea

SuSE

To Set Up and Configure a Suse Repository

1. If you have used a Delinea repository before, it's recommended that you first delete the old repositories in the /etc/zypp/repos.d/*centrify* directory.
2. Create a new SUSE repository.

   You can manually create the repository or you can use a setup script to create the repository automatically.

   > **Note**: For the baseurl parameter, enter your Delinea repository URL token in place of URLTOKEN.

   - **To create the SUSE repository configuration file manually**

     Create a file with the following:

     ```
     /etc/zypp/repos.d # cat /etc/zypp/repos.d/centrify-rpm-suse.repo
     [centrify-rpm-suse]
     name=centrify-rpm-suse
     enabled=1
     autorefresh=1
     baseurl=https://cloudrepo.centrify.com/URLTOKEN/rpm-suse/rpm/any-distro/any-version/$basearch
     type=rpm-md
     repo_gpcheck=1
     gpgcheck=1
     gpgkey=https://downloads.centrify.com/products/RPM-GPG-KEY-centrify
     ```

- **To create the SUSE repository configuration file automatically from a script**

  ```
  curl -1sLf 'https://cloudrepo.centrify.com/URLTOKEN/rpm-suse/cfg/setup/bash.rpm.sh' | sudo -E bash
  ```

3. Refresh the cache.

   ```
   /etc/zypp/repos.d # zypper refresh
   ```

4. Verify the connection to the repository.

   ```
   /etc/zypp/repos.d # zypper packages |grep centrify
   ```

   You should see output similar to the following:

   ```
   mysusemachine:/etc/zypp/repos.d # zypper refresh
   Retrieving repository 'centrify-rpm-suse' metadata ----------[\]


   Retrieving repository 'centrify-rpm-suse' metadata .............[done]
   Building repository 'centrify-rpm-suse' cache .....................[done]
   All repositories have been refreshed.
   utsles15ppcle:/etc/zypp/repos.d # zypper packages | grep Centrify
   | centrify-rpm-suse  | CentrifyDA            | 3.7.0-172        | ppc64le
   | centrify-rpm-suse  | CentrifyDA            | 3.6.1-324        | ppc64le
   | centrify-rpm-suse  | CentrifyDC            | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC            | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-cifsidmap  | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-cifsidmap  | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-curl       | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-curl       | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-ldapproxy  | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-ldapproxy  | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-nis        | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-nis        | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openldap   | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openldap   | 5.6.1-330        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openssh    | 8.2p1-5.7.0.207  | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openssh    | 7.9p1-5.6.1.329  | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openssl    | 5.7.0-207        | ppc64le
   | centrify-rpm-suse  | CentrifyDC-openssl    | 5.6.1-330        | ppc64le
   ```

5. Install the Server Suite Agent rpm package.

   ```
   # zypper install CentrifyDC
   ```

   > **Note**: To uninstall the Server Suite Agent rpm file, you can use the remove command. For example:

   ```
   # zypper remove CentrifyDC
   ```

Delinea

Debian Ubuntu

To set up and configure a Debian or Ubuntu repository

1. Create the repository:

   You can manually create the repository or you can use a setup script to create the repository automatically.

   - **_To create the Debian or Ubuntu repository configuration file manually**

     1. Update the /etc/apt/sources.list file to include the official Delinea package repository.

        ```
        deb https://cloudrepo.centrify.com/URLTOKEN/deb/deb/debian any-version main /etc/apt/sources.list.d/centrify-deb.list
        ```

2. Import your GPG key and update the repository.

   ```
   # bash -c 'wget -O - https://support.delinea.com/s/product-downloads/products/RPM-GPG-KEY-centrify | apt-key add -'
   ```

3. Comment out the no-debsig line in /etc/dpkg/dpkg.cfg to enable GPG signature validation.

   ```
   # grep no-debsig /etc/dpkg/dpkg.cfg
   # no-debsig
   ```

4. Clean and update the local archives.

   ```
   # apt-get clean
   # apt-get update
   ```

- **To create the Debian or Ubuntu repository configuration file automatically from a script**

  ```
  curl -1sLf 'https://cloudrepo.centrify.com/URLTOKEN/deb/cfg/setup/bash.deb.sh' | sudo -E bash
  ```

  Note: Enter your Delinea repository URL token in place of URLTOKEN.

If you manually created your APT repository, the configuration details are in the /etc/apt/sources.list file. If you used the setup script to create the APT repository, the configuration details are in a separate file such as centrify-deb.list in the /etc/apt/sources.list.d directory.

2. Execute the apt list command to verify the repository connection. You should see output similar to the following.

   ```
   # apt list --all-versions | grep centrify
   centrifyda/buster 3.7.0-172 amd64
   centrifyda/buster 3.6.1-324 amd64
   centrifydc-adbindproxy/buster 5.7.0-217 amd64
   centrifydc-adbindproxy/buster 5.6.1-334 amd64
   centrifydc-cifsidmap/buster 5.7.0-207 amd64
   centrifydc-cifsidmap/buster 5.6.1-330 amd64
   centrifydc-curl/buster 5.7.0-207 amd64
   centrifydc-curl/buster 5.6.1-330 amd64
   centrifydc-ldapproxy/buster 5.7.0-207 amd64
   centrifydc-ldapproxy/buster 5.6.1-330 amd64
   centrifydc-nis/buster 5.7.0-207 amd64
   centrifydc-nis/buster 5.6.1-330 amd64
   centrifydc-openldap/buster 5.7.0-207 amd64
   centrifydc-openldap/buster 5.6.1-330 amd64
   centrifydc-openssh/buster 8.2p1-5.7.0.207 amd64
   centrifydc-openssh/buster 7.9p1-5.6.1.329 amd64
   centrifydc-openssl/buster 5.7.0-207 amd64
   centrifydc-openssl/buster 5.6.1-330 amd64
   centrifydc/buster 5.7.0-207 amd64
   centrifydc/buster 5.6.1-330 amd64
   ```

3. Execute the apt install or apt-get install commands to install Delinea packages. For example:

   ```
   # apt install centrifydc centrifydc-nis
   # apt-get install centrifydc-5.7.0-207
   ```

   Note: To uninstall the Server Suite Agent for *NIX rpm, you can use the remove command to delete the Server Suite Agent for *NIX package or the purge command to also delete any configuration files. For example:

```
# apt remove centrifydc=5.7.0-207
```

To Access a Raw Package (wget) Repository for Atomic

Use the WGET repository for Atomic packages or any raw or plain files such as *.zip, *.tar, *.tgz, and so forth .

1. Browse the package index to determine which file you want to download. You can view the package index in HTML or JSON:
    - HTML version is at https://cloudrepo.centrify.com/URLTOKEN/wget/raw/
    - JSON version is at https://cloudrepo.centrify.com/URLTOKEN/wget/raw/index.json

        Note: Enter your Centrify repository URL token in place of URLTOKEN.
2. Download the desired file using either curl or wget. For example:

    curl -1 -O 'https://cloudrepo.centrify.com/URLTOKEN/wget/raw/versions/Latest/CentrifyDC-atomic.x86_64.tgz'

    or

    wget 'https://cloudrepo.centrify.com/URLTOKEN/wget/raw/versions/Latest/CentrifyDC-atomic.x86_64.tgz'

To Set Up and Configure an Alpine Linux Repository

1. To configure the repository automatically, run the following commands:

```
sudo apk add --no-cache bash
curl -1sLf \ 'https://cloudrepo.centrify.com/URLTOKEN/apk/setup.alpine.sh' \ I sudo -E bash
```

   Note: Enter your Delinea repository URL token in place of URLTOKEN.

2. Or, if you want to manually configure the repository, run the following commands:

```
curl -1sLf 'https://cloudrepo.centrify.com/URLTOKEN/apk/rsa.5DD8742729E6E4B2.key' > /etc/apk/keys/apk@centrify-5DD8742729E6E4B2.rsa.pub
curl -1sLf 'https://cloudrepo.centrify.com/URLTOKEN/apk/config.alpine.txt?distro=alpine&codename=v3.13' >> /etc/apk/repositories
apk update
```

When configuring the repository manually, you reference the Delinea public RSA key: apk@centrify-5DD8742729E6E4B2.rsa.pub.

1. Execute the apk add command to install the Server Suite packages. For example:

```
# apk add centrifydc centrifydc-nis
```

   To uninstall the Server Suite Agent for *NIX rpm, you can use the del command to delete the Server Suite Agent for *NIX package. For example:

```
# apk del centrifydc=5.8.0-xxx
```

About the Files And Directories Installed on the Agent

When you complete the installation, the local computer will be updated with the following directories and files for the core Server Suite Agent for *NIX:

/etc/centrifydc	The agent configuration file and the Kerberos configuration file.
/usr/share/centrifydc	Kerberos-related files and service library files used by the Centrify Agent to enable group policy and authentication and authorization services.
/usr/sbin /usr/bin	Command line programs to perform Active Directory tasks, such as join the domain and change a user password.
/var/centrify	Directories for temporary and common files that can be used by the agent.
/var/centrifydc	Before joining the domain, the directory contains basic information about the environment, such as the IP address of the DNS server and whether you installed licensed or express agent features. After you join the domain, several files are added to this directory to record information about the Active Directory domain the computer is joined to, the Active Directory site the computer is part of, and other details.

Depending on the components you select during installation, additional files and directories might be installed or updated. For example, if you install Enterprise Edition, the computer is updated with additional files and directories for auditing.

Joining an Active Directory Domain at a Later Time

At this point, you have delivered the software to target computers, but not changed their configuration. Users still have exactly the same access as they did before installing Server Suite software. The computer's configuration changes only happen when the computer joins an Active Directory domain, that is, joining the domain is what "activates" Server Suite software.

You have the option to automatically join an Active Directory domain when you install Server Suite Agents the install.sh script. In most cases, however, you should not do so unless you have already planned your user migration and created your initial zones. Typically, it is best to analyze the user population and prepare for migration before joining the domain to ensure minimal disruption of user activity and ease the transition to new software. Over time, as you become more familiar with the migration process and refine your zone design, you can adapt the steps to suit your organization.

If you want to join the domain at the same time you deploy the Server Suite software, you should do the following before you install files on the UNIX computers:

1. Download the Server Suite software for all platforms or the subset of platforms you intend to support.
2. Analyze existing user and group accounts.
3. Identify your zone requirements and create the initial zone design.
4. Migrate users and groups into the appropriate zones and role assignments.
5. Use the install.sh script or a custom script to install Server Suite Agents and join the domain.

The additional steps are described in the next sections. You can also manually join a domain at any time after installation by using the `adjoin` command.

Installing the Agent on Solaris Systems

This section covers information about installing the Server Suite Agent for *NIX on Solaris systems. The procedures differ depending on whether you're installing svr4 or IPS packages. If you're installing IPS packages onto a system with Solaris 11 child zones, there's a separate procedure for that deployment.

For which packaged file to use for installation, refer to the table below.

Solaris 10	svr4	x86	centrify-server-suite-2021.1-sol10-x86.tgz
Solaris 10	svr4	Sparc	centrify-server-suite-2021.1-sol10-sparc.tgz
Solaris 11	svr4	x86	centrify-server-suite-2021.1-sol10-x86.tgz
Solaris 11	svr4	Sparc	centrify-server-suite-2021.1-sol10-sparc.tgz
Solaris 11	IPS	x86	centrify-server-suite-2021.1-sol11-i386.tgz
Solaris 11	IPS	Sparc	centrify-server-suite-2021.1-sol11-sparc.tgz

Installing the Solaris Svr4 Agent Packages

Download the solaris package appropriate for your Solaris system and run the install.sh script as mentioned in the "Install interactively on a computer" section. You can follow the same procedure if you're installing on a system with or without child zones.

You can run the following command to verify the Solaris agent package svr4 installation status:

```
pkginfo | grep -i centrify
```

Note that there is no space between "pkg" and "info"; if you search for "pkg info" you'll be searching for IPS packages.

Installing the Solaris IPS Agent Packages

This procedure is for systems where you are doing a fresh install of Server Suite software onto a Solaris 11 system with IPS support.

This procedure is the same as for the regular install script, except that you run the `install-ips.sh` script, not the install.sh script -- see the "Install interactively on a computer" section.

You'll need the centrify-infrastructure-services-VERSION-sol11-i386.tgz file or the `centrify-infrastructure-services-VERSION-sol11-sparc.tgz` file for this procedure, depending on the type of system you have.

To install the Solaris IPS packages

1. Download the Server Suite package for your version of Solaris.
2. Extract the Server Suite packages onto the system.
3. Run the `install-ips.sh` script. For example:

   ```
   ./install-ips.sh
   ```

4. Follow the prompts displayed to select the services you want to install and the tasks you want to perform. For example, you can choose whether you want to:
   - Perform a default installation.
   - Perform a custom installation by selecting the specific packages to install.
   - Join a domain automatically at the conclusion of the installation.

     Depending on your selections, you may need to provide additional information, such as the user name and password for joining the domain.

5. You can run the following command to verify the Solaris agent package IPS installation status:

   ```
   pkg info l grep -i centrify
   ```

   Note the space between "pkg" and "info"; if you search for "pkginfo" you'll be searching for svr4 packages.

Installing the Solaris IPS Agent Packages With Child Zones

When you install the agent onto a Solaris 11 computer enabled with IPS that also has one or more child zones configured, you need to import the agent packages into a new repository and then install directly from the repository.

You do this install in the global zone and the repository will automatically install the files into the child zones.

You'll need the centrify-infrastructure-services-VERSION-sol11-i386.tgz file or the centrify-infrastructure-services-VERSION-sol11-sparc.tgz file for this procedure, depending on the type of system you have.

To install the Solaris IPS agent packages onto a system with one or more child zones

1. Create a directory and extract the IPS tgz file into that directory.

   For example, create directory called "install-ips" and extract the contents of the tgz file into that directory.

   1. Create a repository:

      For example, run the following command to create a repository called "my-repo":

      Tip: You can run the zfs list command to list all of the zone file systems.

      ```
      zfs create rpool/export/my-repo

      zfs set atime=off rpool/export/my-repo

      pkgrepo create /export/my-repo

      pkgrepo set -s /export/my-repo publisher/prefix=centrify

      pkgrepo -s /export/my-repo refresh

      pkgrepo -s /export/my-repo info

      pkg set-publisher -G '*' -M '*' -g /export/my-repo centrify

      pkg publisher
      ```

      You should see the repository listed.

      ```
      PUBLISHER TYPE STATUS P LOCATION
      centrify origin online F file:///export/my-repo
      ```

2. Import the packages into the repository. You need to import the packages that end with .p5p and you need to import them one at a time.

   1. In the directory where you extracted the Server Suite Agent packages, list out the files in that directory (use the ls command).

      The full package list of files that you need to import into the repository looks something like this:

      ```
      centrifyda-3.7.0-sol11-i386.p5p

      centrifydc-5.7.0-sol11-i386.p55

      centrifydc-curl-5.7.0-sol11-i386.p5p

      centrifydc-ldapproxy-5.7.0-sol11-i386.p5p

      centrifydc-nis-5.7.0-sol11-i386.p5p

      centrifydc-openldap-5.7.0-sol11-i386.p5p

      centrifydc-openssh-5.7.0-sol11-i386.p5p

      centrifydc-openssl-5.7.0-sol11-i386.p5p
      ```

   2. Import each package into the repository.

      For example, if you're installing all 8 packages, you'll run the following 8 commands:

```
pkgrecv -s centrifyda-3.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-5.7.0-sol11-i386.p55 -d /export/my-repo '*'
pkgrecv -s centrifydc-curl-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-ldapproxy-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-nis-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-openldap-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-openssh-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
pkgrecv -s centrifydc-openssl-5.7.0-sol11-i386.p5p -d /export/my-repo '*'
```

3. You can verify that the packages imported correctly by listing out the repository packages.

   For example, run the following command:

   ```
   pkgrepo list -s /export/my-repo
   ```

   You'll see a list of packages where each package has a long version. For example:

   ```
   centrify security/centrifydc 5.7.0.207:20200726T052946Z
   ```

3. Install the packages from the repository into the parent and child zones.

   Be sure to reference the package's entire version. When you install the centrifydc package, the other packages for cURL, OpenLDAP, and OpenSSL are also installed.

   For example, to install centrifydc, you'd run the following command, :

   ```
   pkg install -r security/centrifydc@5.7.0.207:20200726T052946Z
   ```

   For example, to install all the packages with one command, you'd run something like this:

   ```
   pkg install -r security/centrifydc@5.7.0.207:20200726T052946Z security/centrifydc-ldapproxy@5.7.0.207:20200726T053320Z security/centrifydc-nis@5.7.0.207:20200726T053352Z
   security/centrifydc-openssh@5.7.0.207:20200727T065442Z security/centrifyda@3.7.0.171:20200725T014652Z
   ```

4. You can run the following command to verify the Solaris agent package IPS installation status:

   ```
   pkg info | grep -i centrify
   ```

   Note the space between "pkg" and "info"; if you search for "pkginfo" you'll be searching for svr4 packages.

Uninstalling the Agent on Solaris Systems

To uninstall the Solaris svr4 packages

1. In the directory where you have downloaded and extracted the Centrify Agent packages, run the following command:

   ```
   ./install.sh -e -n
   ```

2. You can run the following command to verify the Solaris agent package svr4 installation status:

   ```
   pkginfo | grep -i centrify
   ```

   Note that there is no space between "pkg" and "info"; if you search for "pkg info" you'll be searching for IPS packages.

To uninstall the Solaris IPS packages

1. In the directory where you have downloaded and extracted the Server Suite Agent packages, run the following command:

   ```
   ./install-ips.sh -e -n
   ```

2. You can run the following command to verify the Solaris agent package IPS installation status:

   ```
   pkg info | grep -i centrify
   ```

   Note the space between "pkg" and "info"; if you search for "pkginfo" you'll be searching for svr4 packages.

To uninstall the Solaris IPS packages on systems with one or more child zones

1. To uninstall a single package from both parent and child zones, run the following command:

   pkg uninstall -r *packagename*

   For example, to uninstall only the CentrifyDA package, run the following command:

   pkg uninstall -r security/centrifyda

   To uninstall more than one package or all installed packages, enter the package names separated by a space. For example:

   pkg uninstall -r security/centrifyda security/centrifydc-curl security/centrifydc-ldapproxy

2. You can run the following command to verify the Solaris agent package IPS installation status:

   pkg info l grep -i centrify

   Note the space between "pkg" and "info"; if you search for "pkginfo" you'll be searching for svr4 packages.

Sun Solaris Installation Notes

This section describes the unique characteristics or known limitations that are specific to using authentication service on a computer with the Solaris operating environment.

Changing the Local User Password on Solaris

On Solaris, the passwd command is designed to update the databases listed in the nsswitch.conf file or the specific repositories you indicate with the -r option. Therefore, by default, you can use passwd command without any command line options to update your password wherever necessary.

Once you install authentication service and join the domain, however, Active Directory becomes the primary repository for user account information and changing the password for any local user account you need to maintain outside of Active Directory requires you to explicitly specify the repository to update with the -r option.

For example, if you want to change the password for a local user account in /etc/passwd, you must specify the files repository when you run the passwd command:

passwd -r files user

If you want to update the password for an Active Directory user account, you can use the passwd command without the repository option on Solaris 10. For example:

passwd adusername

If you are using an earlier version of the Solaris operating environment, however, you must use the adpasswd command that is installed with authentication service to update the password for Active Directory user accounts. For information about using adpasswd, see the adpasswd man page or the Administrator's Guide for Linux and UNIX.

Installing Authentication Service Packages into Solaris 10 Zones

All zones should be up and running during an upgrade from a previous release of Server Suite Authentication Service and its add-on packages (for example sudo or Server Suite for Web Applications) should not be installed directly into a sparse zone, they should be installed from the global zone only.

Installing Authentication Service Packages into Solaris 11 Child Zones

You need to install SVR4 packaging tools in the child zone before authentication service can be installed.

To check if the SVR4 package has been installed, run

$ pkg info svr4

If it is not installed yet, run the following to install it:

$ pkg install pkg:/package/svr4

Note that the command above may need internet connection (depends on how the IPS repository is configured in the zone).

Creating a Home Directory for New Users on Solaris

In most operating environments, when new users log on successfully, the authentication service will automatically create the user's home directory. On Solaris, however, the home directory is typically auto-mounted over NFS, so the option to automatically create a new home directory for new users is off by default. You can turn on this feature, if suitable to your environment, by adding the following to /etc/centrifydc/centrifydc.conf:

```
pam.create.homedir: true
```

With this flag, the first time a user logs in the home directory will be created. The user will see the message "Failed to create home directory", but this can be ignored.

In Express mode use auto.schema.homedir to specify the home directory for users. Use % as a placeholder for a user's name.

For example:

```
auto.schema.homedir: /export/home/${user}
```

Planning to Use Server Suite Zones

One of the most important aspects of managing computers with Server Suite software is the ability to organize computers, users, and groups into **zones**. This section discusses the primary reasons for using zones and provides an overview of how to analyze and migrate an existing user population into zones, including an introduction to assigning roles that enable users to access computer resources. These topics will then be described in greater detail in the next sections to help you create an initial zone structure and migrate users and groups for a target set of computers.

Why Use Zones?

A zone is similar to an Active Directory organizational unit (OU) or NIS domain. Zones allow you to organize the computers in your organization in meaningful ways to simplify the transition to Active Directory and the migration of user and group information from existing identity stores. The primary benefits to using zones are:

- Identity management through user and group profile definitions
- Access and authorization control through rights and role definitions
- Delegated computer management for zone-based administrative tasks

Zones also enable you to centrally manage configuration policies for computers and users through group policies, but for most organizations the key considerations for designing a zone structure involve:

- Identity management because zones enable you to migrate from a complex UID space, where a user can have multiple UIDs or different profile attributes on different computers or a single UID might identify different people depending on the computer being used. With zones, you can associate multiple UNIX profiles with a user and identify the correct profile attributes for any user on any given computer.
- Access and authorization management because zones enable you to grant specific rights to users in specific roles on specific computers. By assigning roles, you can control who has access to which computers.
- Delegated computer management because zones enable you to assign specific administrative tasks to specific users or groups on a zone-by-zone basis, allowing you to establish an appropriate separation of duties. With zones, administrators can be given the authority to manage a given set of computers and users without granting them permission to perform actions on computers in other zones or access to other Active Directory objects.

In most organizations, the first goal in designing the zone structure is to migrate users and computers from an existing identity store, such as NIS, NIS+, local files, or LDAP, to Active Directory and to do so with the least possible disruption to user activity, business services, and the existing infrastructure. Over time, you can also use zones to organize computers along departmental, geographical, or functional lines using whatever strategy works best for your organization.

Although Server Suite supports a **workstation mode** that does not require you to create and manage zones—a single Auto Zone is defined instead—most organizations find using zones to be an essential part of their migration to Active Directory. The next sections provide more information about why zones are an important part of the planning process. For more information about using Auto Zone, see Deploying to a single Auto Zone.

Identity Management Using Zones

For most organizations, it is impractical to attempt to rationalize user accounts across the enterprise to achieve a single global UID for each user. For example, most organizations have multiple identity stores already in use on their current UNIX platforms. These identity stores may include LDAP directories, NIS or NIS+ domains, and local /etc/passwd and /etc/group configuration files. With these multiple identity stores, it is common for a single user to have a different user name, UID, group memberships, or other attributes defined for different computers.

Zones allow you to import the information from these legacy identity stores without consolidating the multiple profiles that each user may have. For example, a single user might have an account in a UNIX LDAP directory, another defined in a NIS domain, and one or more local /etc/passwd files. Zones enable the profiles from these different identity stores to map to a single Active Directory user account without changing the user profile defined in each of the legacy directories. By keeping the profiles intact, the user's file ownership and log in permissions are not affected by the migration to Active Directory, making the transition from a legacy system to Active Directory more transparent to end users and less of a management burden for the deployment team.

Role-based Access Control and Zones

As a practical matter, you may choose to use Server Suite zones to ease migration to Active Directory by creating a separate zone for each legacy identity store. However, you can also use zones to group computers by department, by function, or by any other criteria you choose. Using zones in this way gives you a great deal of flexibility in controlling who has access to the UNIX, Linux, and Mac OS X computers in your environment and makes it easier to set up account information for new users based on job function or other criteria.

Through role assignments, zones provide a scope of resources particular users can access, allowing you to define who can do what on which computers. For example, all of the computers in the finance department could be grouped into a single zone called "finance" and the members of that zone could be

restricted to finance employees and senior managers, each with specific rights, such as log on to a database, update certain files, or generate reports. This gives you better control over access to systems based on well-defined roles. You can also limit access to certain types of applications, such as database management utilities or web services. For example, you can define specific actions specific users are allowed to perform by assigning them different roles in different zones.

Using Zones to Delegate Administrative Duties

Zones can also be useful for grouping computers that form a natural administrative set or that should be managed by different administrative teams. For example, you may want to group computers that are managed by a local support organization in one zone and computers that are managed by a corporate IT group in another zone.

Using zones, you can then control what different groups of users can do within the zones they have permission to access. For example, you can set up regional zones to provide a separation of duties, authorizing users in San Francisco to manage computers and user profiles in their local office while a team in Barcelona has authority to join computers and manage group profiles for offices located in Spain.

Zones provide a convenient way for you to assign individual administrative responsibilities to specific users or groups based on a set criteria, such as department, geographic location, or functional role.

Deploying to a Single Auto Zone

In most cases, if you are deploying on Linux or UNIX computers and have an existing user population to migrate to Active Directory, you would create a hierarchical zone structure of multiple zones. However, multiple zones are not required for all situations. You can greatly reduce the time required and complexity of your deployment if a single zone suits your organization's needs. For example, if you are deploying on Mac OS X or Windows computers or if you have a mix of computer platforms but do not have an existing user population to migrate, you might benefit from deploying agents using the Auto Zone option.

With Auto Zone, you have a single zone for an entire forest. All of the users and groups you have defined in Active Directory for the forest automatically become valid users and groups on the computers that join the Auto Zone. If the forest has a two-way trust relationship with another forest, all Active Directory users defined in that trusted forest are also automatically valid on computers that join the Auto Zone.

If you simply want to use the Active Directory users and groups you have already defined on the nonWindows computers you manage, you can skip the planning and creation of zones and simply add computers to the Auto Zone when you join the domain. The UNIX profile attributes that are required to access computers in the Auto Zone are then automatically derived from user attributes in Active Directory or from settings defined in group policies or configuration parameters.

> **Note**: You cannot use Auto Zone to give automatic access to users and groups in a forest or domain with a one-way trust relationship with another forest or domain.

You can use Auto Zone without enabling any group policies or changing any of the default configuration settings. You can also join a domain through the Auto Zone without installing Access Manager. However, you can use group policies or configuration parameters to specify a subset of Active Directory users or groups as valid Auto Zone users. The settings are then enforced on computers in the Auto Zone.

Using Auto Zone can make sense in small or larger organizations if you are not migrating existing users and groups or maintaining legacy UNIX profile attributes. However, if you use Auto Zone, you cannot use zone-specific features. For computers in the Auto Zone, you cannot configure rights and roles, assign roles to users and groups, or provide different profile attributes on different computers.

For information about joining a domain using Auto Zone, see the man page for adjoin or the *Administrator's Guide for Linux and UNIX*. For information about using group policies or configuration parameters, see the *Group Policy Guide* or *Configuration and Tuning Reference Guide*.

Classic and Hierarchical Zones

If you plan to deploy using zones—which is the most common deployment model—you have to option to create **classic** or **hierarchical** zones. Classic zones provide a simple model for organizing computers and backwards compatibility for organizations with older versions of the Server Suite Agent. Hierarchical zones enable you to establish parent-child zone relationships, allowing profile attributes, rights, and roles to be inherited down the zone hierarchy. Classic zones are peers to each other and do not inherit profile attributes, rights, or roles from each other.

One of the first decisions you need to make in planning your zone structure is whether you will use classic zones, hierarchical zones, or some combination of both.

Should You Use Classic Zones?

Classic zones provide a simple structure for delineating users and groups based on a criteria you choose, such as by region or department. They are most

appropriate if you have a well-defined and well-managed UNIX namespace with very few users who require special handling because of multiple profiles or conflicting profile attributes.

Classic zones are simple to manage as long as you only need a few. For example, imagine you have three regional zones with no users in common that are managed independently by their own zone administrators with only one enterprise system administrator who must have a profile in each zone. In that scenario, classic zones provide a simple solution because only one user account, the enterprise system administrator, must have a profile in each zone.

However, classic zones are very limited in complex environments where users need profiles in multiple zones or where there are multiple independently-managed UNIX namespaces to migrate to Active Directory. That is because classic zones do not share data across zone boundaries. The data must be created and managed in each zone independently. By contrast, hierarchical zones support inheritance, enabling you to create parent and child zones that share information as needed. Because classic zones do not support inheritance, you cannot use variables to define profile attributes or any other hierarchical zone features.

For most organizations, classic zones are primarily used to enable a new zone that works with pre-5.0 versions of the Server Suite Agent. If you have an older version of Server Suite software installed and already have some zones deployed in your environment, you can continue to use those zones as-is. After upgrading, you then have the option to create any new zones as classic zones to operate within the legacy zone environment or as hierarchical zones.

> **Note**: If you already have zones deployed, you can convert them to hierarchical zones after you deploy the new version of Server Suite software, if you choose to do so. However, there's no requirement for you to convert existing zones to hierarchical zones.

When Should You Use Hierarchical Zones?

For most organizations, Server Suite recommends that you use hierarchical zones for all new zones that you create. Hierarchical zones provide greater flexibility to inherit profile information, rights and role definitions, and user and group role assignments.

Because hierarchical zones allow you to share or override information at any point in the hierarchy, they also allow you to design a simpler zone structure than classic zones and support an easier deployment model. Typically, a simpler zone structure is easier to manage, but hierarchical zones also allow you to implement a very sophisticated zone structure to address complex access control rules, if you so choose.

How Many Zones Do You Need?

The goal in planning to use zones is to have a fairly small number of zones that organize the computers and users in your organization most effectively. As an example, consider an organization where some UNIX computers are used to host financial applications. Those computers are centrally managed by the IT organization, which follows well-established conventions for issuing user login names, user IDs, and home directories. The same organization has a software development group that includes numerous UNIX workstations that are not centrally managed by the IT organization and computers and accounts are added when needed and managed independently.

Because enterprise-wide conventions are not enforced for the UNIX computers in the software development group, it's possible that the local login names and user IDs may conflict with the names and IDs used on the computers running the financial applications. In addition, users in the software development group may use a different convention for their home directories or prefer different login shells.

Without zones, the IT organization would need to eliminate any duplicate user IDs and verify each login name was unique across all of the computers. By placing the computers running the financial applications in one zone and the computers in the development lab in another zone, the IT organization can avoid the overhead of checking and changing existing account information and can set default zone settings, such as different default home directories or login shells, that are most appropriate to the users in each zone.

There are many different approaches you can take to defining the scope of a zone, including organizing by platform, department, manager, application, geographical location, or how a computer is used. The factors that are most likely to affect your initial zone design, however, will involve migrating user and group profiles, identifying the appropriate access control policies and role assignments, and delegating administrative tasks to the appropriate users and groups. For many organizations, the most important issue during the initial deployment is a successful migration of existing users. Using hierarchical zones with the ability to override attributes simplifies this task, helping to reduce the total number of zones you need to deploy.

A Closer Look at Using Zones in a Hierarchical Model

In older versions of Server Suite software, zones were always parallel with each other and did not share or inherit data except through manual processes. Starting with Infrastructure Services 2012, however, you have the option to use hierarchical zones that support the inheritance of user and group data and provide a great deal of flexibility for defining the rights and roles for who can access which computers and what those users can do on the computers to which they have access.

How Inheritance Provides Additional Benefits

As discussed in Why use zones?, the primary benefits to using zones are:

- Identity management through user and group profile definitions
- Access and authorization control through rights and role definitions
- Delegated computer management for zone-based administrative tasks

Hierarchical zones provide additional flexibility for each of these benefits. For example, because hierarchical zones allow inheritance, hierarchical zones enable you define partial profiles and use variables that can be substituted at run-time when a user accesses a specific computer in a particular zone. Hierarchical zones also enable you to define access control rules and delegate administrative tasks at any point in the zone hierarchy.

This flexibility makes planning for hierarchical zones a key component of a successful deployment.

How Many Levels Should You Use in the Zone Hierarchy?

There are no predefined limits to the number of zones that can be used in a zone hierarchy or the number of levels deep zones can be nested in the hierarchy you define. For practical purposes, however, it is recommended using a hierarchy similar to the following:

- One or more top-level **parent** zones that include basic profile information for all users and groups that access the UNIX, Linux, and Mac OS X computers.
- One to three levels of intermediate **child** zones based on natural access control or administrative boundaries.

At each level in the hierarchy, profile information and access controls are inherited from the zone above and either applied or overridden by the child zone settings. At the lowest level of the hierarchy, you can override profile attributes or role assignments on any individual computers using **machine override** settings, if needed.

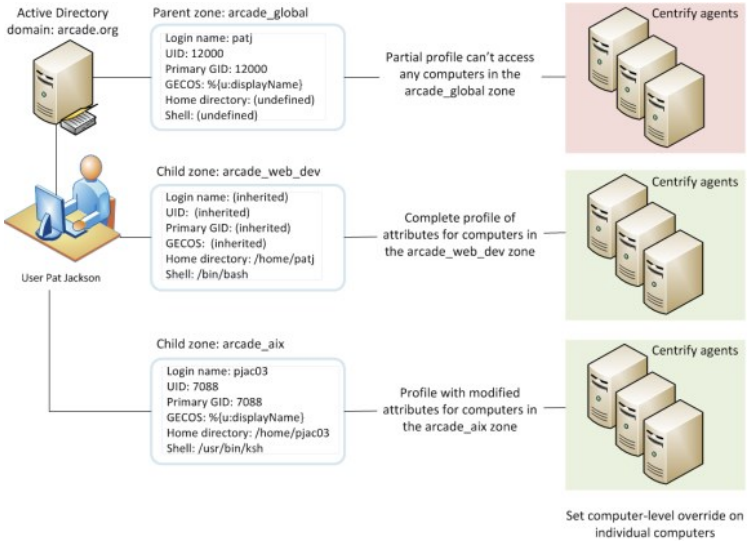
In addition, hierarchical zones support computer-based access rules, called **computer roles**, that enable you to selectively map a set of users with a particular role assignment access to a particular set of computers.

Identity Management and Inherited Profile Information

User and group profiles specify attributes such as the UID, primary group, home directory, and shell that are required for logging on to UNIX computers. You can specify all or part of the profile anywhere in the zone hierarchy, but users must have a complete profile to access computers they have permission to access. If the user or group profile is incomplete, it is invalid and ignored.

Working with Partial Profiles in the Zone Hierarchy

The profile information in the zone hierarchy is resolved from top to bottom for each user. For example, assume the user Pat Jackson has the login name patj and UID 12000 defined in the parent zone arcade_global and those profile settings are inherited without change, along with a default shell, home directory and other properties that are defined in the child zone arcade_web_dev. In a second child zone, arcade_aix, the UID for patj is set to 7088 to override the inherited UID. Changes to the profile properties can be made in any zone and inherited down the tree down to overrides set for specific individual computers, if needed.



Working with Variables in the Zone Hierarchy

Partial profiles enable you define a subset of profile attributes for users and groups that can be completed by lower level zones in the zone hierarchy. You can also define variables for resolving profile attributes. The variables are then substituted at run-time by adclient. For example, adclient can resolve the variable %/% to a platform-specific home directory for each user without having the attribute manually defined. You can set the variables at any level in the zone hierarchy, and they are inherited and resolved, or can be overridden, at a lower level in the tree.

You should note that variables can only be used to define profile attributes in hierarchical zones. You cannot import them or use them in classic zones.

Complete Profiles Do Not Grant Access

Creating user profiles in a zone does not give users access to any computers in the zone. The zone hierarchy simply creates a set of profiles with the potential to be granted access to computers. In previous versions of Server Suite software, enabling a UNIX profile for a user in a zone granted that user access to the computers in that zone by default. With hierarchical zones, the profile information only establishes the required properties for the user's identity, but does not grant access to any computers in any zones.

Access to computers is controlled through the definition of rights and roles.

Access Controls and the Assignment of Rights and Roles

A user must have a complete UNIX profile to log on to any computer in a zone. However, a complete profile alone does not allow a user to access any computers. The user must also have at least one role assignment that grants access somewhere in the zone hierarchy before any type of access is granted. Role assignments can be made anywhere in the zone hierarchy and inherited at a lower level in the tree.

Understanding Roles and Rights

Rights represent specific operations users are allowed to perform. A **role** is a collection of rights that can be defined in a parent or child zone and inherited. For example, a role defined in a parent zone can be used in a child zone, in a computer role, or at the computer level.

There are only a few predefined rights, called system rights. The system rights for Linux, UNIX, and Mac OS X are:

- **Password login and non password (SSO) login are allowed**: Specifies that a user is allowed to log on interactively using a password or without a password using a single sign-on token.
- **Non password (SSO) login is allowed**: Specifies that a user is allowed to log on using a single sign-on token.
- **Account disabled in AD can be used by sudo, cron, etc.**: Specifies that an account that is disabled is allowed to access the computer. This right enables service accounts that run without a password to perform operations.
- **Login with non-Restricted Shell**: Controls whether a user gets a full shell or is forced into a restricted shell. Users must be assigned at least one role with this right to have access to a standard shell environment. A restricted shell only allows a user to execute explicitly defined commands.

The system rights for Windows computers are:

- **Console login is allowed**: Specifies that users are allowed to log on locally using their Active Directory account credentials.
- **Remote login is allowed**: Specifies that users are allowed to log on remotely using their Active Directory account credentials.

In addition to the platform-specific system rights, there is a common system right that allows users to bypass auditing or role restrictions to log on when there are problems on a computer. The **Rescue rights** option allows you to specify the users who can log on if problems with the authorization cache or the auditing service on a computer are preventing all other users from logging on.

You grant users permission to access computers by assigning them to a role that includes one or more access rights. By default, zones only contain the following predefined roles to grant basic access rights:

- **UNIX Login** role allows users assigned this role to log on and access UNIX computers in the zone.
- **Windows Login** role allows users assigned this role to log on and access Windows computers in the zone.

There are additional predefined roles that grant specific rights, such as the right to log on if auditing is required but not available. The predefined roles exist in each zone, but their role names are qualified by the zone name so that the same role name in a parent zone and a child zone are considered different roles. For deployment, the predefined roles enable you to migrate existing users without developing custom role definitions. After deployment, you can define additional rights, roles, and role assignments to refine how users and groups access computers in different zones.

Working with a Candidate Set of Profiles

Ultimately, the purpose of the zone structure is to determine who has access, and what kind of access, to a computer. The candidate set of profiles that have the potential to access a computer is resolved by traversing the zone hierarchy from top to bottom. Because profile data is defined separately from the role assignments that control access, you can define an inclusive set of user profiles in a parent zone to create a candidate set that can then be applied to multiple child zones. In each child zone or at the individual computer level, you can use role assignment to control access for specific users from the inclusive candidate set.

Delegation in Hierarchical Zones

Hierarchical zones enable you to create a separation of duties for zone administration without recreating user and group profiles in multiple child zones. You can create full or partial profiles in the parent zone and inherit them into the child zone. Within each child zone, zone administrators can modify the profiles, as needed, and assign roles to control access to the computers they manage.

Designing a Zone Structure for Your Environment

Because the flexibility of hierarchical zones is a key element in designing the zone structure for your deployment, the next sections describe how to set up and use parent and child zones through sample deployment scenarios. The scenarios illustrate a basic deployment model, which will then be used to discuss how to migrate existing users and groups to Active Directory.

Your own zone structure and deployment model can be more complex than the one described in this guide. However, the deployment model described in the next sections is intended to ensure that you have a successful initial deployment. Over time, it is likely that you will change and adapt the zone structure to requirements that are specific to your organization. There are also multiple ways to accomplish the tasks described in the next sections of this guide. You can use other strategies and techniques for deployment if appropriate for your organization.

Preparing To Migrate Existing Users And Groups

This section describes how to prepare your environment for migration and computer access, including how to create and configure the initial zones. This chapter also describes how to import existing account information into Active Directory, set up provisioning for new users and groups, and configure role-based access controls.

Collecting And Analyzing Users and Groups

Before you create any zones, you should collect and analyze information about existing users and groups in the target set of computers. After you have investigated user and group attributes and identified invalid accounts and conflicting attributes, you can draft a basic zone design that addresses the needs of that user population. The goal of the initial zone design is to provide access to those users who currently have access, so they can transition to Active Directory with no disruption to their work. Later, you can refine access to computers through role assignments, filtering, and other options, if needed.

Collecting Information from Other Departments in Your Organization

Before you look at the content of identity stores you want to migrate, you should consider other sources of information that will help you identify a definitive set of legitimate users. For example, it can be useful to contact people in other departments who have reliable knowledge about the current organization or historical knowledge about how the organization has evolved. Individuals with information about a segment of the user population can help you identify accounts that are obsolete or were created for testing, or belong to users who have left the company or moved to another department.

As a starting point for collecting information about existing users and groups, consider doing the following:

- Contact HR to get an up-to-date list of current active-duty employees, contractors, and consultants. You can use this information to compare personnel records to the UNIX accounts to be migrated. After you identify which accounts correspond to people in the organization, you can create a spreadsheet to record the UNIX user names, UIDs, and other useful fields for those accounts.
- Contact enterprise security administrators or department-level UNIX administrators to determine whether all of the accounts defined for a computer still need access to that computer. For example, you should determine if any users validated as current employees have changed departments or job functions. If a user no longer needs access to some computers, you may not need to add a profile for that user.
- Identify any conventions used in defining the namespace. For example, is there a standard format for the contents of the GECOS field? How do the conventions used for UNIX, Linux, or Mac OS X accounts compare to the conventions used in Active Directory? For example, is the convention used for the UNIX login name the same as the convention used for the user's sAMAcountName in Active Directory? Does the GECOS field follow the same conventions as the user's displayName in Active Directory?
- Identify which user attribute fields that can be used as primary keys for identifying a unique user. Depending on the conventions you use for creating new accounts, the user name, user identifier (UID), or the GECOS field may be a reliable field for identifying real users and mapping them to Active Directory accounts. If you use a standard provisioning convention across platforms for an attribute such as the GECOS field or user name, the convention makes it much easier to identify unique users and map user profiles to Active Directory accounts.

Using a Script to Retrieve User and Group Profiles for Each Computer

Alternatively, you can write a script to retrieve all of the /etc/passwd and /etc/group files in the target set of computers. For example, to create a hostname.passwd and hostname.group file for each computer in a target set of computers, you might use code similar to the following:

```
for name in `cat hostname.txt`; \
do scp $name:/etc/passwd $name.passwd; \
scp $name:/etc/group $name.group; \
done
```

This sample script includes the computer host name in the file name, so that you can determine which user and group definitions came from which computer. If you use a script to collect user and group information, copy all of the files generated by the script to a common location for analysis.

Collecting Data from NIS Domains

If you're migrating users and groups from a NIS domain, you can use ypcat or niscat to generate a copy of the NIS passwd and group maps once for each NIS domain. For example, run commands similar to the following:

```
ypcat passwd > `domainname`.passwd
ypcat group > `domainname`.group
```

If you are collecting user and group information from NIS maps, copy all of the files generated by these commands to a common location for analysis.

Identifying Accounts that Should Not Be Migrated

After you have collected information about the existing users and groups in the target set of computers, examine each of the passwd and group files and NIS domain maps to create a list of users and groups that you do not plan to migrate into Active Directory. For example, in most cases, there's no compelling business reason to migrate default system accounts, such as nfsnobdy or games, that will not map to Active Directory users. You should also eliminate accounts for people who have left your organization, and accounts that are locked or obviously invalid.

Eliminate Default System Accounts

In most cases, you can ignore all UNIX users with a UID less than 99 because those are the default operating system accounts. You may also want to skip migration for some or all UNIX service accounts unless you explicitly want to manage those service accounts, and any privileged commands they run, through Active Directory and zones.

You can manage the passwords for UNIX service accounts using Access Manager without having those accounts defined in zones or in Active Directory. Therefore, you may want to leave most or all of the service accounts as locally defined accounts.

In general, the only reasons to migrate default system or service accounts to Active Directory are:

- If you want to use Active Directory password policies for the account.
- If the service account itself owns one or more privileged commands that you want to manage through Centrify role definitions rather than locally in the sudoers file.

Typically, only service accounts that own special permissions, such the oracle user account, are migrated to Active Directory.

Remove Other Invalid Accounts

You should also skip migration for users who have left the organization and profiles that contain invalid data. Scan the data set for user accounts and groups that are locked or indicate they were created for testing. You should also check for profiles that contain obvious errors or legacy information no longer used.

In some cases, you may need to contact workstation or application owners directly to determine whether a profile should be skipped for migration. For example, assume the /etc/group file contains an entry for clowns with krusty, bozo, and tadams as members and there are valid user profiles for those three users. You may suspect the clowns group was created for some local testing, and therefore, not a candidate for migration. However, there's no definitive indication that the clowns group should be skipped without more information.

Create a List of the Users and Groups to Ignore

Add all of the accounts that should not be migrated to a text file. For example, create a text file named user.ignore and include all of the user accounts you don't want to migrate into Active Directory. For default system and service accounts that have known UIDs, you can create the text file programmatically using code similar to the following:

```
cat *.passwd | \egrep "x?:[0-9]:[0-9].|x?:[0-9][0-9]:[0-9].|x?:60[0-9][0-9][0-9]:[0-9]|x?:65[0-9][0-9][0-9]:[0-9]" | \
cut -d ":" -f1 | \
sort | \
uniq | \
sed 's/^/\^/' > user.ignore
```

Other accounts you have identified as invalid can be added manually or using code if they share some common attribute, such as LOCKED in the password field.

Analyze User Profiles for Conflicting Attributes

After the initial analysis to remove profiles that should not be migrated, you have a candidate data set of users and groups to import. The next step is to analyze the attributes in user profiles to identify any potential problems that you will need to address when you move the profiles into zones. Delinea Professional Services offers scripts that assist in this process. If you are analyzing the files manually or writing your own scripts, here are the common issues you need to check for:

- Determine whether any user name has more than one UID in the target set of computers. The UID is the primary way of determining file ownership and file permissions. On a single UNIX system, a user can only have one UID. However, across all of the computers in the target set, the same user name might have more than one UID.
- Determine whether any UIDs are associated with more than one user name.
- Determine whether any users have other profile conflicts, such as more than one primary GID, home directory, or shell on the computers in the target

set.

In doing your preliminary analysis, keep in mind that you need to know which user profiles are associated with which people in your organization. For example, do the user names ldavis and davle refer to the same person (Len Davis) or to different people (Len Davis and Leslie Davidson).

This analysis of existing user profiles will help you identify the requirements of your initial zone design. The zone design allows you to use conflicting attributes as-is, without modifying any of the legacy data. You need to be aware of where there are conflicts so you can address them, but you do not need to change values for any attributes.

Analyze Group Profiles for Conflicting Attributes

You need to perform a similar analysis across the groups in the target set of computers. Delinea Professional Services offers scripts that assist in this process. If you are analyzing the files manually or writing your own scripts, here are the common issues you need to check for:

- Determine whether any group name has more than one GID in the target set of computers. Like the UID, the GID affects file ownership and file permissions. On a single UNIX computer, a group name can only have one GID. However, across all of the computers in the target set, the same group name might have more than one GID.
- Determine whether any GIDs are associated with more than one group name.
- Determine whether any group has a different set of members on any computers in the target set. Group membership is particularly important for zone design. The members of a group must be consistent across all of the computers in a zone.

As with the user analysis, this analysis of existing group profiles will help you identify the requirements of your initial zone design. The zone design allows you to use conflicting attributes as-is, without modifying any of the legacy data. You need to be aware of where there are conflicts so you can address them, but you do not need to change values for any attributes.

Create a Working Set of User and Group Profiles

After you have identified profiles that should not be migrated and noted any conflicting attributes you need to address, you have a known set of user and group profiles that you plan to migrate into Active Directory. The next step is to remove the users who should be ignored from list of users to import, and to remove the groups that should be ignored from the list of groups to import. You can do this manually or write a script that removes the profiles defined in the user.ignore and group.ignore files and outputs the results to a new file. For example, you might use code similar to the following to remove ignored users and generate a working set of user profiles:

```
mkdir output; \
for name in `cat hostname.txt`; \
do egrep -v -f user.ignore $name.passwd > \output/$name.passwd; \
done
```

How Migration Affects the Zone Design

As discussed in Why use zones?, identity management is one of the primary benefits of using zones. Identity management is important because most organizations have an existing user population where users can have multiple UIDs or other attributes, such as different default shells, on different computers and groups with the same name can have different members. Each user has one Active Directory user object but may have multiple UNIX profiles, some with attributes in common and some with different settings. Zones allow you to migrate the profile information as it is defined, setting overrides where necessary, so that you can manage and report on the accounts without rationalizing the user namespace.

For all of the computers in a zone, a user or group has one profile definition, but the user or group could have different profile attributes on the computers in a different zone. Hierarchical zones make the zone design even more flexible. Hierarchical zones allow you to define one or more profile attributes in a parent zone and use those profile attributes in all child zones. In practice, this enables you to define a dominant set of attributes in a parent zone, and inherit the common attributes in one or more child zones. You can also override any attribute at any point in the zone hierarchy down to an individual computer.

For example, if a UNIX administrator has a consistent profile across all of the UNIX computers, but a customized home directory on two Mac OS X computers, you could define the default profile for the user in a parent zone, then create a child zone for the Mac OS X computers and inherit all of the profile attributes except the home directory setting.

In planning the migration, you identify the attributes that are the same across the target set of computers and where there are differences. If you use hierarchical zones, the primary task is identify one or more potential parent zones. For example, if you are migrating two NIS domains, you might create two parent zones because the UID space is unique in each domain but there would be conflicting attributes if the domains were combined into a single parent zone. The computers in each of the parent zones would inherit the UID and other profile attributes from each distinct NIS domain.

After you have identified one or more parent zones, you can plan how you will use child zones and overrides to manage profile attributes, implement access

controls, and delegate administrative duties.

Creating the First Zone

It is recommended that you plan to use hierarchical zones and create at least one top-level parent zone. A single top-level zone for your organization is also useful for long-term management of UNIX profiles. You can have more than one top-level parent zone. For example, if your organization has subsidiaries that are run independently or distinct geographical locations managed by different teams, you may want to create separate parent zones for those lines of business or locations.

Having a single top-level parent zone enables you to create an administrative group of super-users who can log on to every UNIX computer in your organization. It also allows to define some common rights and roles that can be inherited by child zones and the computers in those zones. Having a global or master zone for the entire organization also simplifies setting up provisioning for new accounts. However, there's no restriction on the number of parent or child zones you create. If you have a distributed environment and delegate administrative authority to separate teams, you can create as many parent zones as you find useful.

This guide describes how to set up the migration environment using one top-level parent zone. If you create more than one parent zone, you may need to repeat steps or extrapolate additional steps from the information presented here.

Create a Top-level Parent Zone

Before you migrate users and groups or add computers to the domain, you must have at least one zone. It is recommended that you create one top-level parent for your organization, which is similar to having a single forest root domain.

To create the top-level parent zone

1. Log on to the Windows computer where Authentication & Privilege is installed and open Access Manager.

   If you are not currently connected to the appropriate forest, specify the domain controller to which you want to connect.

2. In the console tree, select **Zones** and right-click, then click **Create New Zone**.

3. Type the zone name and, optionally, a longer description of the zone.

   In most cases, you should use the default parent container and container type that you created when you configured the Active Directory forest, and the default zone type, which creates the new zone as a hierarchical zone, then click **Next**.

   The only reasons for changing the default settings would be if you want to:

   - Create a zone in a new location to separate administrative activity for different groups of administrators.
   - Create zones as organizational units because you want to assign group policy objects to zones.
   - Create a classic zone for backwards compatibility or are using the Microsoft Services for UNIX (SFU) schema.

     For additional details about any of the zone fields, press F1 to view context-sensitive help.

4. Review the information about the zone you are creating, then click **Finish**.

Add Provisioning Groups to the Parent Zone

The next step in configuring the top-level parent zone is to create two Active Directory groups that will enable automated provisioning and de-provisioning of users and groups in the toplevel parent zone. By creating these provisioning groups in the parent zone, you can integrate the provisioning of UNIX users and groups with your existing processes for provisioning Active Directory users.

The provisioning groups are not required for migration, but a recommended configuration for the top-level parent zone you are creating as the first zone in the environment.

It is recommended you follow the naming conventions suggested for these groups. If you use a different naming convention, you should be sure it is well documented in your internal process documentation.

To add the provisioning groups for user and group profiles to the parent zone

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Select Provisioning Groups, right-click, then select **New > Group**.
4. Type the group name using the format ZoneName_Zone_Groups. For example, if the zone name is arcadeGlobal, type arcadeGlobal_Zone_Groups, then click **OK**.

   The Zone Provisioning Agent will use this group when processing the business rules for adding or removing group profiles in the parent zone.
5. Select Provisioning Groups, right-click, then select **New > Group**.
6. Type the group name using the format ZoneName_Zone_Users. For example, if the zone name is arcadeGlobal, type arcadeGlobal_Zone_Users, then click **OK**.

   The Zone Provisioning Agent will use this group when processing the business rules for adding or removing user profiles in the parent zone.

To prevent problems in UIDs and GIDs for existing users and groups, you should import existing user and group profiles before defining the business rules for automated provisioning of new accounts. After you complete the migration of the existing user population, you will define the business rules for the ZoneName_Zone_Groups and ZoneName_Zone_Users groups you just created.

Create Groups for the Default Roles in fhe Parent Zone

The next step in configuring the top-level parent zone is to create two Active Directory groups for the default listed and UNIX Login roles that are predefined in hierarchical zones.

- If you have a single top-level parent zone, users with a listed role can be recognized as having a valid profile on every UNIX computer in the organization. However, users in the listed role are not allowed to log on to any of those computers.
- If you have a single top-level parent zone, users with a UNIX Login role can log on to every UNIX computer in the organization.

For the toplevel parent zone, the UNIX Login role is intended for enterprise-level systems administrators who need to be able to log on to any UNIX computer in the organization. Because these are powerful roles in the parent zone, only a limited number of users would ever be assigned to these roles. However, the listed and UNIX Login roles are key components of migration when you create one or more child zones. If no users in the organization will be assigned these roles in the parent zone, you can skip the creation of the Active Directory groups for roles in the parent zone.

To create the groups for listed and UNIX Login roles in the parent zone

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Select the User Roles organizational unit, right-click, then select **New > Group**.
4. Type the group name using the format ZoneName_Role_RoleName. For example, if the zone name is arcadeGlobal, type arcadeGlobal_Role_Listed, then click **OK**.
5. Select the User Roles organizational unit, right-click, then select **New > Group**.
6. Type the group name using the format ZoneName_Role_RoleName. For example, if the zone name is arcadeGlobal, type arcadeGlobal_Role_Login, then click **OK**.

Delegate Administrative Tasks on the Parent Zone

The next step in configuring the top-level parent zone is to delegate administrative authority to the Zone Administrators group and to delegate specific permissions to the service account for the Zone Provisioning Agent to enable automated provisioning of user and group profiles in the parent zone.

To delegate administrative tasks on the top-level parent zone

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the top-level parent zone, right-click, then click **Delegate Zone Control**.
4. Click **Add**.

5. Change the Find list from User to **Group**, type z, then click **Find Now**.
6. Select Zone Administrators in the results, then click **OK**.
7. Click **Next**.
8. Select **All** to enable members of the Zone Administrators group to perform all administrative tasks on the top-level parent zone, then click **Next**.
9. If you are delegating the task of joining computers to a zone, you can specify the scope of computers you can join to the zone; you pick a container in Active Directory to grant access to.

   If you leave the scope blank, the scope is the domain root. Be aware that the postalAddress field is used for information about joining computers to a zone; if you lookup the permissions for people you've delegated the task of joining computers to a zone, they'll have permissions to the postalAddress field for the affected computers.
10. Review your selections, then click **Finish**.
11. Right-click, then click **Delegate Zone Control**.
12. Click **Add**.
13. Type all or part of the service account name for the Zone Provisioning Agent that you created in About Zone Provisioning Agent and its requirements, click **Find Now**, then select the service account in the results and click **OK**.
14. Click **Next**.
15. Select the following delegation rights for the Zone Provisioning Agent service account, then click **Next**:
    - Change zone properties
    - Add users
    - Add groups
    - Remove users
    - Remove groups
16. Review your selections, then click **Finish** to save the changes and close the dialog.

Link a Role Group to a Role Assignment in the Parent Zone

The next step in configuring the top-level parent zone is to link the Active Directory role groups created in Create groups for the default roles in the parent zone with the listed and UNIX Login role definitions that are predefined in the parent zone. You create this link between an Active Directory group name and the combination of rights associated with a role name by assigning the Active Directory group to the role.

1. Start Access Manager.
2. In the console tree, expand **Zones**, the top-level parent zone, and Authorization nodes.
3. Select Role Assignments, right-click, then click **Assign Role**.
4. Find the ZoneName_Role_Listed Active Directory group, then click **OK**.
5. Click **Browse**.
6. Select the listed role from the list of available roles, then click **OK**.
7. Check that the Start immediately and Never expire options are selected and appropriate or deselect those options and set start and end times, then click **OK**.
8. Repeat Step 3 through Step 7for the ZoneName_Role_Login Active Directory group and the UNIX Login role.

Create One or More Child Zones

After you have created a parent zone and prepared it with provisioning and role groups, you can create one or more child zones. You can create the child zones based on any logical model you choose. This is where the analysis of common and conflicting attributes and some creativity come into play.

Logical Models for Defining Zones

Because the zone design uses hierarchical zones, you can override attributes at any zone or computer level to deal with conflicts in legacy profile data. With this flexibility, you can experiment with different possible designs, for example, based on delegated administrative authority or physical location. Some common models for grouping a set of computers, users, groups, roles, and rights in the same zone include:

- **By shared identity store** For example, existing identity stores, such as NIS domains or a centralized user and group database, often provide a natural boundary for zones. This strategy is especially effective if each identity store has a consistent namespace, without profile conflicts. It is less effective if the computers that share a common administrative group use local /etc/passwd and /etc/group file to store account information.
- **By application or function** For example, you might want to groups all of your database servers or web farm servers into their own zones. As part of this design, you might need to evaluate whether you are combining development computers with production servers and what role assignments you'll need to control what users can do on each type of computer.
- **By geographical region or line of business** For example, all of the UNIX computers that support a business unit could be logically grouped together. As part of this design, you might evaluate whether different business units should be responsible for provisioning users or assigning roles within their own business unit.
- **By host name** If you already have a meaningful host name convention that identifies machine owners or primary function, you may want to create zones based on that naming convention.
- **By platform and operating system** You can use this strategy, for example, to create separate zones for Red Hat Linux workstations and Sun Solaris UNIX workstations.
- **By department or user community** You can use this strategy, for example, to create separate zones for the computers that host financial applications and computers used by software developers.

You are not required to create child zones. You could control access to the parent zone through role assignments. For most organizations, however, one or more child zones makes it easier to assign roles and manage group membership.

Depending on your target set of computers, you may decide to start with one or two child zones or skip the creation of a child zone.

Create a Child Zone under the Parent Zone

Creating a child zone is similar to creating a parent zone. You select the parent in the left pane, then create and configure the child zone to prepare an environment into which you will migrate existing users and groups.

To create a child zone under the parent zone

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the top-level parent zone, right-click, then click **Create Child Zone**.
4. Type a name and description for the child zone, then click **Next**.

   For example, if you are organizing by functional group, this zone might be finance or engineering. If you are organizing by data center location, the child zone might be sanfrancisco or seattle.
5. Click **Finish** to complete the zone creation.

The new zone is listed under the Child Zones node in the left pane.

Create Role Groups for Child Zones

The next step in configuring the child zone is to create two Active Directory groups for the default listed and UNIX Login roles that apply to this zone.

- In the child zone, users with a listed role can be recognized as having a valid profile but only on computers that are joined to the child zone. Users in the listed role for the child zone cannot log on to any of the computers joined to the child zone.
- In the child zone, users with a UNIX Login role are allowed to log on to every UNIX computer joined to the child zone if they have a UNIX profile for the zone.

For the child zone, the UNIX Login role is intended zone-level administrators and users who were previously able to log on to the UNIX computers joined to the child zone. The listed and UNIX Login roles are key components of migration when you create one or more child zones.

To create the role groups for listed and UNIX Login roles in the parent zone

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Select User Roles, right-click, then select **New > Group**.
4. Type the group name using the format ChildZoneName_Role_RoleName. For example, if the child zone name is sanfrancisco, type

sanfrancisco_Role_Listed, then click **OK**.
5. Select User Roles, right-click, then select **New > Group**.
6. Type the group name using the format ChildZoneName_Role_RoleName. For example, if the zone name is sanfrancisco, type sanfrancisco_Role_Login, then click **OK**.

Delegate Administrative Tasks on the Child Zone

The next step in configuring the child zone is to delegate administrative authority to the Zone Administrators group. The steps are the same for the child zone as the parent zone, except that you expand the Child Zones node and select the name of the child zone before selecting the Delegate Zone Control command. You should still assign the Zone Administrators group All permissions.

You also have the option of assigning the permissions to join or leave to the Join Operators Active Directory group. If you pre-create computer accounts and allow the computer to join itself to the Active Directory domain, you can skip this step.

If you don't want to pre-create the computer account and allow the self-service join, you must give members of the Join Operators group the following administrative tasks:

- Join Computers to the Zone
- Remove Computers from the Zone
- Modify Computer Profiles

Link Role Groups to Role Assignments in the Child Zone

The next step in configuring the child zone is to link the Active Directory role groups created in Create role groups for child zones with the listed and UNIX Login role definitions that are predefined in the child zone. You create this link between an Active Directory group name and the combination of rights associated with a role name by assigning the Active Directory group to the role. The steps are the same for the child zone as the parent zone, except that you expand the **Child Zones** node and select the name of the child zone before selecting the **Authorization** node.

When you search for the Active Directory group to assign, you will select the ChildZoneName_Role_Listed, for example sanfrancisco_Role_Listed, for the listed role, and ChildZoneName_Role_Login, for example sanfrancisco_Role_Login, for the UNIX Login role.

Users who are added to the ChildZoneName_Role_Login group will be able to log on to computers that are joined to the child zone or any of its own children, but will not be able to log on to computers in other child zones.

Create Computer Objects For The Target Set Of Computers

When you manage UNIX computers with Centrify software, you add computer objects to Active Directory for those computers. These computer objects can be created automatically when a computer joins the domain, or created in Active Directory before the computer joins the domain. In most cases, Centrify recommends that you create the computer account objects before joining, if possible.

For deployment and migration, creating the computer objects before joining provides the following key advantages:

- You can define computer-level overrides before computers are added to the zone. This allows you to resolve issues with divergent UNIX profiles without having to change file permissions at the file system level.
- You can check who will have access to which UNIX computers before those computers join the Active Directory domain.

Pre-creating the computer objects enables you to check that you have user profiles and role assignments correctly defined before you join the UNIX computers to zones. Verifying this information before the join operation helps to ensure a smooth migration without disrupting users' access to files or applications.

Prepare A Computer Object Before Joining

In most cases, you should pre-create the computer object for every UNIX computer in every zone. For individual computers, you can use the Prepare Computer wizard to guide you through the process. However, you will probably want to create a Windows or UNIX script for performing the operation repeatedly. For example, you can use adedit or the Windows API to create a script.

To prepare a computer account in Active Directory using Prepare Computer

1. Start Access Manager.
2. In the console tree, expand the **Child Zones** node, then expand the child zone for this computer to join.

3. Select the Computers node, right-click, then click **Prepare Computer**.
4. Accept the default preparation options, then click **Next**.
5. Accept the default to **Create a new computer object**, then click **Next**.
6. Type the name of the computer object to create and modify the DNS host name of the computer object, if necessary.

   The computer name is the name of the computer principal in Active Directory. The DNS name is how the UNIX computer is currently registered in DNS. If you have a disjointed DNS namespace, you should be sure the DNS name is the name used in the computer's DNS entry.
7. Click **Change** and navigate to the organizational unit for storing computer principals. For example, if you created the organizational unit structure described in Creating recommended organizational units, select UNIX Servers and Workstations and click **OK**, then click **Next**.
8. Select an option for joining the computer to the domain, then click **Next**.
   - If you want to require users to interactively join the computer to Active Directory, click **Browse** to select the Join Operators group.
   - If you want to allow the computer to join itself to the zone, select **Allow the computer to join itself to the zone**. This option automatically associates the computer with the correct zone, so there's less chance of a human error.
9. Click **Browse** to select the Zone Administrators group, then click **Next**.

   With this setting, users in the Zone Administrators can override any inherited attributes of a UNIX user or a UNIX group profile on the computer.
10. Review your selections, then click **Next** to create the computer account.
11. Click **Finish** to complete the process.

You have now finished preparing the environment for migration and are ready to begin importing groups and users and assigning them appropriate roles.

Migrating Existing Users To Hierarchical Zones

Now that you understand how zones are used and have prepared an environment for an initial migration, you are ready to import the existing users and groups that you have identified as candidates for being migrated to Active Directory.

This section uses a sample data set to illustrate how to migrate an existing user population into hierarchical zones and how to assign the appropriate roles to convert from a legacy authentication model to Active Directory and Server Suite.

Importing Group Profiles

After you have created one or more zones and separated the users and groups to ignore from the users and groups that you think should be migrated to Active Directory, you must decide which groups apply to which zones. For example, if you have some groups with the same group profile and group membership in all zones, you would import those groups into the top-level parent zone so that they also exist, with the same definition, in all child zones. If a group is only applicable for computers in a child zone, you can import the group profiles directly into that zone. You can also override group profile attributes on specific computers, if needed.

After you have made these decisions, importing the groups is a simple process using either the **Import from UNIX** wizard or ADedit scripts with two important considerations:

- Group names must be unique in Active Directory. If you create a group with a common name, such as admins, you cannot create another group with the same name.
- Having the same UNIX group name on computers in different zones can create group collisions and inadvertent privilege escalation or file ownership conflicts.

To prevent group name collisions, Centrify recommends that you include the zone name in the Active Directory group name. You may also want to add a suffix that identifies the group as an UNIX security group. In most cases, you create the Active Directory group object for the UNIX group in the UNIX Groups organizational unit if you created the organizational unit structure described in Creating recommended organizational units.

You should import group profiles and create the corresponding Active Directory groups for those groups before you import users. If you import group profiles first, you can resolve secondary group membership for users immediately after you import user profiles.

Import Unix Groups that Apply to All Computers into the Parent Zone

If your organization has a default UNIX administrators group or security group that you want to be available on all UNIX computers, that group is a good candidate for importing into the parent zone. Other groups that might be candidates for the parent zone are special purpose UNIX groups that own sudoers permissions that apply to all UNIX computers or an auditing group that requires access to all computers.

If you have identified any common groups, use the Import from UNIX wizard or a script to import the UNIX groups that should be available for all computers into the top-level parent zone.

To Import Unix Groups Using the Import from Unix Wizard

1. Start Access Manager.
2. In the console tree, expand **Zones** and the top-level parent zone.
3. Select UNIX Data, right-click, then click **Import from UNIX**.
4. Click **UNIX configuration files**, then click **Browse** to locate and select the group file to import, then click **Next**.
5. Select the option to automatically shorten the UNIX name, if desired, then click **Next**.
6. Leave **Store in Active Directory** selected and click **Next**.
7. Select **Check data conflicts while importing**, then click **Finish**.

   This step places the profiles under Groups as Pending Import.
8. Select one or more group names that are Pending Import, right-click, then select **Create new AD groups**.
9. Click **Browse**, navigate to the UNIX Groups organizational unit and click **OK**, then click **Next**.

10. Click **Add a prefix to group name**, type the parent zone name and an underscore (*), select the group scope as Global, then click **Next**. For example, if the parent zone name is arcadeGlobal, use the prefix arcadeGlobal.*

    Optionally, click **Add a suffix to group name** and type a suffix that identifies the group as a UNIX security group, for example, _unix.

11. Review the information displayed, then click **Finish**.

For more information about importing groups, see the *Administrator's Guide for Linux and UNIX*.

Import Unix Groups that Apply Only to a Specific Zone into a Child Zone

From your initial analysis and zone design, you should also have a reasonable plan for groups that apply to specific child zones. Groups imported into a child zone are visible to all the UNIX computers in that zone, but not in other zones. For example, assume you have identified an application-specific group, ora_app01, that allows users to use a database, and the database application exists three computers. In your zone design, you decide those three computers should be a single child zone. In that case, you import the ora_app01 group profile into the child zone group because the database application group is only relevant to the UNIX computers in the child zone.

The steps for importing into a child zone are the same as for the parent zone, except that you select the UNIX Data under the child zone name in the console tree and specify the child zone name as the prefix for the Active Directory group name.

Import a Group Profile or Override Attributes on Specific Computers

In some cases, you may have a UNIX group that only exists on one computer in a zone or exists on more than one computer but has different attributes on different computers. You can use computer-level overrides to handle these cases. Computer-level overrides enable Zone Administrators to create and manage group profile attributes manually for individual computers.

To create a group profile for a specific computer

1. Use Active Directory Users and Computers to create an Active Directory group in the UNIX Groups organizational unit. If the group only applies to a specific computer, you may want to use the computer name as the prefix.
2. Start Access Manager.
3. Expand the console tree to display the individual computer object under the zone the computer will join.
4. Expand UNIX Data, select Groups and right-click, then click **Create UNIX Group**.
5. Click the attributes to define, type the appropriate values, then click **OK**.
    - Click **GID** to manually specify a GID for the group profile on the selected computer.
    - Click **UNIX group name** to manually specify a group name for the profile on the selected computer.

Avoiding Group Collisions When Using Computer-level Overrides

If you create group profile overrides on individual computers, you should make sure that the UNIX group name and GID are not being used by any other groups in the parent or child zone. If the group profile defined for the computer is the same as a group profile defined for a group in the parent or child zone, users who should only be able to access files on the local computer may be able to access files owned by the group defined for the parent or child zone. This can be a difficult problem to identify. For example, assume you have an Active Directory group named contract_admins, but you have used the UNIX group name admins and the same GID as a group in the parent zone. Any user who is a member of the contract_admins group in Active Directory is going to have the same GID as the parent zone's admins group. If that happens, members of the contract_admins group will have access to the same files as the admins group in the parent zone.

The only way to identify when this problem occurs is by running the following command for a user in the contract_admins group:

id -a

Importing User Profiles

You can import user profiles into the parent zone or into child zones. If you import user profiles into the parent zone, all existing users will be included in the candidate set of users who have the potential to log on to all of the UNIX computers in the organization. However, they are not granted any access by default. Instead, the management of identity information, such as the user name, UID, and primary group, is separate from privilege management. Users cannot access any UNIX computers until they are assigned a valid role with the specific permissions they need to be recognized, allowed to log on, or run specific commands.

Although you can import users into the parent zone without granting them access rights, you may prefer to import them into one or more child zones. By

importing users into specific child zones, you can limit the scope of their potential access. In general, this option is applicable for the majority of your end-users and can apply to other users, such as database administrators, project managers, and contractors who won't ever need access to all the UNIX computers in the organization.

At this stage you should decide whether to give users the potential to access all computers in the organization, or only the computers in one or more specific child zones. After you import the user profiles, you will use the default listed and UNIX Login roles or custom roles to control access to the UNIX computers.

The steps for importing user profiles into a parent or child zone are essentially the same as importing groups. You can use the **Import from UNIX** wizard or ADedit scripts to import the profiles into one or more zones. The profile information for any user can be different in each zone. If the profile information is divergent on any computer within a zone, you can set computer-specific overrides for any or all attributes.

Tip: If you are importing users from a file, you can write a script that modifies the GECOS field to use the same format used in the Active Directory displayName attribute before importing so that users are automatically mapped to their corresponding Active Directory accounts. For example, if your convention for the GECOS field is first_name last_name (Jae Wilson) but the convention used in Active Directory is last_name, first_name (Wilson, Jae), you must manually map the UNIX user account to the Active Directory account. If you modify the format of the GECOS field before importing, the Import from UNIX wizard can automatically suggest a candidate for mapping the UNIX user to an Active Directory user, if an account exists.

After you import users, their profiles are placed under Users as Pending Import. If the user has an existing Active Directory account, you can select the user name, right-click, then select **Extend existing AD user**. If an Active Directory account does not exist, you can select the user name, right-click, then select **Create new AD users**. You can then use **Check Status** to resolve group membership for Pending Groups. This command adds the imported users to the appropriate Active Directory groups that have UNIX profiles in the zone to complete the first phase of the migration to Active Directory.

For more information about importing users and resolving group membership, see the *Administrator's Guide for Linux and UNIX*.

How Group Membership Works Within Zones

When a UNIX group profile is imported into a zone, its group name and GID are recognized by all computers joined to that zone. However, the group membership might vary by computer. For a user to be a member of the UNIX group, the user must:

- Be a member of the Active Directory group.
- Have a complete UNIX user profile defined somewhere in the zone hierarchy (in the parent zone, a child zone, or with computer-level overrides).
- Be assigned the listed role or the UNIX Login role somewhere in the zone hierarchy (in the parent zone, a child zone, or with computer-level overrides).

For example, assume the users Alison and Clyde are assigned the UNIX Login role for the Engineering zone. As discussed in Create role groups for child zones, that means they are also listed as members of the Engineering_Role_Login role group in Active Directory. Clyde is also a member of the denali project group in the Engineering zone and has a profile defined in the parent zone. Alison's profile is defined in the Engineering zone. If the denali project group (Engineering_Denali in Active Directory) is added to the Engineering zone, both Alison and Clyde can log on to computers in the Engineering zone, but only Clyde will be a member of the denali UNIX group in the Engineering zone.

Assigning Roles to Existing Users and Groups

You have now imported the existing user and group profiles for a target set of computers into Active Directory. This is one critical component of migration because users must have a valid UNIX profile, that is, a unique user name, UID, primary GID, home directory and shell, in a zone for them to be recognized as valid users. However, Server Suite separates UNIX profile management from UNIX privilege management. Users cannot log on to UNIX computers until they are assigned a role that allows them to log on to those computers.

As discussed in Access controls and the assignment of rights and roles, a role is a collection of rights and there are two default roles: the listed role and the UNIX Login role. As part of deploying Server Suite software with the least disruption to your environment, your existing users must be able to log on to the UNIX computers they currently use. That is the primary purpose of the UNIX Login role: to allow you to quickly give log on access to a set of users in one or more zones. The UNIX Login role in the parent zone is intended for enterprise administrators who need log on access to all computers. The UNIX Login role in the finance zone would be for those users who currently have interactive access to the limited number of computers in that zone and would expect to have that access after migration.

The listed role is intended for users who need a valid profile defined but do not need interactive log on access to the computers in a zone. For example, you assign the listed role to remote NFS users so that they have access to their files without the ability to log on and open a shell. You can also use the listed role to give users access to applications, such as ClearCase or Samba, that require a UNIX profile without the ability to log on locally or remotely. The listed role in the software-dev zone would be for those ClearCase users who need to be recognized on all computers in the zone so they can check files in and out.

The next step in the migration is to identify which users should be assigned to each role in each zone you have created.

Using Active Directory Groups for Roles

For most organizations, the most efficient way to manage role assignment is by adding users to Active Directory groups, then managing those groups. Therefore, for management purposes, a Centrify access role should always be linked to an Active Directory security group. The Active Directory groups that identify the users in specific Centrify user roles are stored in the User Roles organizational unit. All of the users in a specific role group will share a common set of rights under UNIX. You can then use machine-level overrides for handling edge cases for individual computers.

Adding Users to Role Groups

There are many different ways you can add UNIX user profiles to an Active Directory group. For example, you can manually select a UNIX profile in a zone, right-click, then select **Add to a group** or select groups under the Role Assignments node for the zone, and modify the group membership. In most organizations, however, you can leverage your existing provisioning process. If your current provisioning process involves managing a group in Active Directory, whether it is through automated scripts or human processes, you can use the same process for provisioning UNIX users.

Migrating Existing Users into The Unix Login Role in The Parent Zone

In Create groups for the default roles in the parent zone, you created Active Directory security groups for UNIX Login and listed roles in the parent zone. If you want to give all users the potential to log in to all UNIX systems, you can make them members of the parentZone_Role_Login group.

Users who are members of this group and have a complete UNIX profile in the parent zone can log on to all UNIX computers that are joined to the parent zone and all UNIX computers joined to the child zones of the parent zone. However, if you add users to the parentZone_Role_Login group in Active Directory, but do not define a UNIX user profile in the parent zone, those users will only be able to log on to the UNIX computers in the child zones where they have a UNIX user profile defined or the individual computers where you define machine-level overrides to give them a UNIX profile.

The default UNIX Login role associated with the parentZone_Role_Login group does not grant any additional privileges. It simply allows users to log on to UNIX computers. Therefore, one strategy for migrating users is to add them all to parent zone's Login role group. You can then control access based on where the user's UNIX profile is defined and control what the user can do using additional role assignments. For example, you may create custom roles to grant expanded UNIX privileges.

Migrating Existing Users into the Unix Login Role in Child Zones

If you define user profiles for most of your users in the parent zone, you should not make them members the parentZone_Role_Login group. Instead, you can add users to the appropriate childZone_Role_Login groups. All of your existing UNIX users who can currently log on interactively to existing UNIX systems should be added to one or more childZone_Role_Login groups. For example, users who currently have access to all of the computers in the Engineering zone should be added to the Engineering_Role_Login Active Directory group. If those users also have a UNIX profile in the parent zone or the Engineering zone, they will be able to log on to all of the computers in the Engineering zone. If a user only needs access to a specific computer in the zone, you can use a machine-level override to give the user access to that specific computer.

You can use the Access Manager console, Active Directory Users and Computers, ADEdit or custom scripts to add UNIX user profiles to the appropriate childZone_Role_Login groups. If possible, you should integrate this part of the migration with your existing provisioning process to ensure that future requests for UNIX role assignments use the processes that line of business personnel already understand.

Migrating Existing Users into the Listed Role in Child Zones

After you have assigned users who must be able to log on to the UNIX Login role, you should identify users who should be assigned the listed role to limit the number of users allowed to log on. The listed role is intended for existing UNIX users who have a UNIX user profile in one or more zones that you want to allow to be listed in getent output without the ability to log on to UNIX computers in those zones.

The listed role is most commonly used for users who access UNIX applications, such as ClearCase, or Samba, or an NFS-mounted file system, that require a UNIX profile. In practical terms, however, this role also allows you to migrate users you aren't sure have been authorized for access. With this role, the user profile is recognized but the user cannot log on locally or remotely.

You can use the Access Manager console, Active Directory Users and Computers, ADEdit or custom scripts to add the UNIX user profiles to the appropriate childZone_Role_Listed groups. If possible, you should integrate this part of the migration with your existing provisioning process to ensure that future requests for UNIX role assignments use the processes that line of business personnel already understand.

Keep in mind that the childZone_Role_Listed group affects all the UNIX computers joined to the specified child zone. Before you move a user to the childZone_Role_Listed group, you should check whether there are any computers in the zone that the user must be able to access to prevent accidentally locking the user out. You can use a machine-level override to grant the UNIX Login role on a specific computer, if needed.

Using a Computer-level Override for the Unix Login Role

You can also create computer-level overrides for the UNIX Login or listed role, if needed. This is not typically part of the migration process. However, if your initial analysis identified a zone where overrides would be useful, you can include overrides in your migration plan. For example, assume you have a zone where most of the user profiles are common across a set of computers. If you import the UNIX profiles for that user population, you see that two users would have access to a UNIX computer where they previously did not have access. To preserve the existing access while migrating from the legacy environment, you can define the UNIX profile in the zone but control access for those two users with a computer-level override.

Managing Role Assignment Without Role Groups

You are not required to use Active Directory security groups to manage role assignments. You can manually add users and groups to roles within any zone. Manually adding a user or group to a role without using Active Directory groups makes integration with provisioning systems more difficult, however. Most identity management and provisioning systems are designed to work with Active Directory groups inherently. Therefore, associating Active Directory groups with Server Suite roles typically provides easier integration with existing provisioning processes.

If you decide to manually manage role assignments, you can use the Server Suite Access Module for Windows PowerShell, Server Suite Access SDK, or ADedit to create scripts that manipulate the objects in Active Directory. Role assignments are stored in Active Directory using Microsoft Authorization Manager containers. If you want to add and remove user and group assignments, you will need to develop custom code to accomplish those tasks.

Verifying Effective Users On Each Zone

Now that you have imported profiles and assigned existing users to the appropriate roles, you can verify who has access to the computers in each zone before you proceed with joining a domain. Checking the Effective Users in each zone enables you to verify the users who have been assigned the UNIX Login and listed roles before any users are affected by the changes.

You should have a checklist of the users who require interactive access on the computers in the target set and which user profiles you suspect only need to be recognized without the ability to log on. You can then using the Effective Users option to see the role assignments for the pre-created computer objects in the target set of computers. By comparing the list of users to the role assignments, you should be confident that you are ready to complete the migration by joining UNIX computers to the Active Directory domain.

Performing this step before joining the domain helps to ensure the transition to Active Directory does not interfere with end-users daily work or the delivery of business services. Therefore, verifying UNIX Login and listed access before joining computers to the domain is a key part of a successful migration.

To access the Effective Users for a zone

1. Start Access Manager.
2. In the console tree, expand **Zones** and the top-level parent zone.
3. Select a zone, right-click, then click **Show Effective UNIX User Rights**.
4. Review the list of UNIX user profiles for the zone in the UNIX users section.
5. Select a user name to display additional information about each user:
    - **Zone Profile** displays details about inherited profile attributes. For existing users being migrated, the profile attributes are typically explicitly defined. If a profile is defined higher up in the zone hierarchy, the Inheritance tab indicates where the profile attributes are defined.
    - **Role Assignments** lists the role assignments for the selected user in the zone. For the initial migration, users must be assigned the UNIX Login or listed role.
    - **PAM Access** lists the specific PAM application access rights associated with the roles a user is assigned. For example, the default UNIX Login role has the login-all PAM access right, which enables PAM authentication for all computers in the zone.
    - **Commands** lists the specific UNIX command rights associated with the roles a user is assigned. For example, you can define a role that allows users to run specific privileged commands as root. You can click the Commands tab to see the specific privileged commands defined for the role.
    - **SSH Rights** lists the specific secure shell (ssh) command rights associated with the roles a user is assigned.
6. Click **Close** when you have finished checking role assignments for the users in target computer of computers.

You can also select **Show Effective UNIX User Rights** for individual UNIX computers and generate Hierarchical Zone reports that describe the effective rights for computers and users.

Adding Existing Users and Groups to Provisioning Groups

After you have added the existing users and groups to the appropriate Login and Listed role groups, the next step for completing the migration to Active Directory is to add the existing user and group profiles to the Provisioning Groups you created for the parent zone. This step is not directly related to data migration, but enables you to prepare the environment for automated user and group fulfillment using on the Zone Provisioning Agent.

Add Existing Users To The Provisioning Group For The Parent Zone

At this point, you have imported legacy data into one or more child zones and accepted divergent profile attributes using computer-level overrides. You should now add all of your imported UNIX users to the provisioning group in the top-level parent zone. Adding users as members of the provisioning group will enable the Zone Provisioning Agent to define a new "universal" UNIX profile for legacy users based on business rules you establish for the parent zone. The new profile will not affect the existing file ownership, but will make it easier to provision and deprovision users moving forward.

As discussed in Installing Zone Provisioning Agent, the Zone Provisioning Agent enables you to define business rules for creating new UNIX profiles for new UNIX users. After you complete the migration and enable the Zone Provisioning Agent, it runs at a regularly scheduled interval to determine whether there are new users or users who should be removed. At each interval, the Zone Provisioning Agent compares the members of the parent zone's Users provisioning group with the user profiles currently defined for the zone.

If there are UNIX profiles for users who aren't members of the provisioning group, the Zone Provisioning Agent removes those user profiles. To prevent the Zone Provisioning Agent from removing the imported data, you must add the Active Directory users associated with the imported user profiles to the parent zone's Users provisioning group.

To add existing UNIX users to the provisioning group for the parent zone

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Expand the Provisioning Groups organizational unit, then select the parentZoneName_Zone_Users group. For example, if the parent zone is arcadeGlobal, select arcadeGlobal_Zone_Users, right-click, then select **Properties**.
4. Click the **Members** tab, then click **Add**.
5. Search for and select the imported user accounts that you have mapped to Active Directory users, then click **OK**.
6. Click **OK** to save the provisioning group and close the Properties.

Add Existing Groups to the Provisioning Group for the Parent Zone

As with imported users, you should also add all of your imported UNIX groups to the provisioning group in the top-level parent zone. Adding the group profiles as members of the top-level provisioning group will enable the Zone Provisioning Agent to define a new "universal" UNIX profile for each group based on business rules you establish for the parent zone. The new profile will not affect the existing file ownership, but will make it easier to provision and deprovision users moving forward. Adding the UNIX group profiles to the top-level parent zone ensures that the Zone Provisioning Agent does not remove the imported groups from the zone.

To add existing UNIX groups to the provisioning group for the parent zone

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Expand the Provisioning Groups organizational unit, then select the parentZoneName_Zone_Groups group. For example, if the parent zone is arcadeGlobal, select arcadeGlobal_Zone_Groups, right-click, then select **Properties**.
4. Click the **Members** tab, then click **Add**.
5. Search for and select the imported user accounts that you have mapped to Active Directory users, then click **OK**.
6. Click **OK** to save the provisioning group and close the **Properties**.

Joining Computers to a Domain and Zone

You have completed the preparation of the environment and added existing users and groups to Active Directory. The steps up to this point have not affected the day-to-day activities of any UNIX users or groups, and have not changed the configuration of any UNIX computers. The final step in the migration requires you to join UNIX computers to the Active Directory domain. This step does have the potential to affect end-users.

This section describes how to complete the migration by joining the target set of computers to an Active Directory domain and a Server Suite zone.

Using Adjoin on New Computers

You can run the adjoin command interactively or in a script to join UNIX computers to Active Directory. One advantage to using the adjoin command is that it enables you to add the join operation to the steps for building a new UNIX computer. For example, if you have a process for provisioning a new UNIX computer, you can add an adjoin step that allows the new UNIX computer to join itself to Active Directory. Provisioning new computers to join the domain when they are built ensures that there are no new local users being defined on those UNIX computers.

Running Adjoin Requires Unix and Active Directory Privileges

On UNIX, running adjoin requires you to log on as root, be a member of the wheel group, or have root equivalent privileges in the sudoers file. On Mac OS X computers, adjoin requires the administrator account and password.

Specifying the Required Options

The basic syntax for the adjoin command is:

adjoin [*options*] domain_name [--zone zone_name | --workstation]

The domain_name should be a fully-qualified domain name; for example, sales.acme.com. If you are using adjoin to provision new computers, there are several options you should specify on the command line or in the script.

- Use the --container or -c option to specify the location for the computer account. Typically, you should use the organizational unit that you created for UNIX Servers and Workstation under the top-level UNIX organizational unit. It must be the location you used when you pre-created the computer object. For example:

  -c "ou=UNIX Server and Workstations,ou=UNIX"

- Use the --selfserve or -S option to specify that you want the computer to join itself to the Active Directory domain.
- Use the --zone or -*z* option to specify the name of the zone to join. You must specify a zone name unless you are joining Auto Zone using the --workstation option.
- If you have a disjointed DNS environment where the Active Directory domain for the computer account does not match the name of the DNS domain, you must also specify the -name and
  --alias options. The --name option specifies the name of the Active Directory computer object and the --alias will be the fully-qualified DNS name of the computer.

For example, update your provisioning process for a new computer to include a command similar to the following:

adjoin -c "ou=UNIX Server and Workstations,ou=UNIX" -S -z production arcade.net

For complete information about adjoin options, see the adjoin man page.

Pre-staging Before Using Adjoin on a New Machine

When joining a large AD environment, the join procedure can take a very long time - up to dozens of minutes. This becomes a concern in some use cases, such as starting an Amazon EC2 instance that needs to join the domain to provide service.

To speed up the adjoin process, the adjoin --prestage option uses existing cache files instead of populating cache from scratch.

Some preparation is required to take advantage of the --prestage option:

- Prepare a pre-staged cache directory on a joined machine
- Copy the cache directory to the new machine

Security Requirements

To use the --prestage option, ensure the following:

- Joined and new machine requirements:
  - The --prestage option can only be used between machines that have the same platform, architecture, and Authentication Service (Centrify DirectControl) release version installed.
  - Adclient cache data encryption feature cannot be enabled on the joined machine. See the adclient.cache.encrypt parameter.
- Pre-staged cache directory on joined machine requirements:
  - On a joined machine, create or designate a directory for the pre-staging cache files.
    - The directory must be in a safe path. That means all levels of parent directories are owned by system accounts.
    - The directory cannot be either group or world writable.
- Content for the pre-staged cache directory on the joined machine:
  - Place the cache files (dz.cache, dc.cache, gc.cache,.idx and kset. files) in the specified directory.
  - Ensure the cache files are owned by system accounts.
  - Files cannot be either group or world writable.
  - Symlink is not allowed for the cache files.
- Zone hierarchy changes are not allowed between the staging directory and the new machine. This includes:
  - zone name change
  - zone GUID change
  - zone schema change

Preparing to Use the --prestage Option

1. Create a directory on a joined machine. For example, /pre.
2. Stop adclient on that machine.
3. Copy the /var/centrifydc/ directory to the pre-staged directory on the joined machine.

   For example:

   Copying the /var/centrifydc/ directory to the pre-staged directory, /pre, places a copy of the required files in /pre/centrifydc/.
4. Verify the pre-staged directory on the joined machine contains all the .idx, .cache, and kset. files.
5. Copy the pre-staged directory to the new machine.

   Use a method of your choice, such as scp or sftp.

   This is done so the pre-staged files are available locally on the new machine.
6. Add the option to the adjoin command when adding the new machine. The syntax is:

   -E | --prestage <directory>

   where directory is the path to the pre-staged directory on the new machine.

   For example, if the pre-staged files are in directory, /pre/centrifydc/, use the following adjoin command.

   adjoin -z <zone> -E /pre/centrifydc<domain>

Verify Authentication After Joining the Domain By Logging On

As the final step in the initial migration, you should verify that authentication for an Active Directory user is successful. You can do this by logging on to the UNIX console using either the UNIX user name or the Active Directory User Principal Name for a user assigned to the UNIX Login role. When prompted, type the Active Directory password for the account. If you are able to log on using the Active Directory password, you know that authentication is being handled by Active Directory and the user account has been successfully migrated.

You should also verify that you can log on remotely using a secure shell (ssh) connection and that you can use other services such as ftp.

If users have trouble logging on after a UNIX computer has joined the domain, it is typically because they're not assigned the UNIX Login role or don't have a

valid UNIX profile in the zone. You can use the Show Effective UNIX User Rights command to check which users have profiles and what roles have been assigned to users who have access to the selected computer.

Provisioning New User and Group Profiles After Migration

After you have completed the basic migration for a set of existing users and groups, you can continue with the Centrify deployment by configuring the environment for automated provisioning of new users and groups. At this stage, you have already built the foundation for the automated addition and removal of users and groups. The next steps involve defining the business rules for creating new user and group profiles. The goal of this section is to help you identify and integrate a provisioning process for new UNIX users and groups.

Integrating with Existing Provisioning Processes

The Zone Provisioning Agent and the provisioning groups you created in Add provisioning groups to the parent zone are intended to integrate the provisioning of UNIX users and groups with your existing account fulfillment process. Those groups enable you to leverage existing processes because most organizations have well-defined and standardized procedures for provisioning new Active Directory users based on Active Directory group membership.

If possible, you would like to use the same or a similar process for provisioning UNIX users and groups. If you can integrate the provisioning of UNIX users and groups with your existing process, the people in your organization can use tools they are familiar with and won't have to learn an entirely new process.

However, defining the business rules for adding new user and group profiles to zones requires some planning. In particular, you need to make decisions about Active Directory group membership, primary group definitions for users in zones, and how profile attributes are defined.

Defining the Business Rules for New Groups in the Parent Zone

You have already started the process of integrating the provisioning for UNIX users and groups when you added imported accounts to the Active Directory provisioning groups in Adding existing users and groups to Provisioning Groups. The next step is to define the business rules for creating new UNIX group profiles in the top-level parent zone.

> **Note**: The business rules you define only affect new UNIX user and group profiles. The imported legacy data remains unchanged, and the Zone Provisioning Agent will not modify any attributes on the existing user and group profiles.

Configure the Business Rules for Automated Provisioning of Group Profiles

You configure the business rules for automated provisioning of group profiles on a zone-by-zone basis. When you use hierarchical zones, you typically want to configure the business rules for the parent zone so that the profile can be inherited in all child zones. Remember that the profile, by itself, does not provide any access to the computers in the child zones, and that you can override any inherited attributes in any zone or on individual computers.

To Configure the Business Rules For Groups in the Parent Zone

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the top-level parent zone, right-click, then click **Properties**.
4. Click the **Provisioning** tab.

   If you are defining business rules for a parent hierarchical zone and want to establish a "source zone" for profile attributes, click **Advanced**. You can then select the Source zone for any or all user and group profile attributes. If you select Source zone for any attribute on the Advanced Provisioning page, you can click **Browse** to search for and select the zone to use as the source zone. In most cases, selecting a source zone is not necessary if you using hierarchical zones, but this option can be useful if you are migrating from classic to hierarchical zones.
5. Click **Enable auto-provisioning for group profiles**.
6. Click the Find icon to search for and select the "groups" zone provisioning group as the Source Group.

   If you followed the recommended naming convention, search for and select parentZoneName_Zone_Groups. For example, if the zone name is arcadeGlobal, select arcadeGlobal_Zone_Groups.
7. Select a method for assigning a new GID to new UNIX group profiles:
   - **Generate from group SID** generates new GIDs that are guaranteed to be unique in the forest based on the Active Directory security identifier (SID) of the group. Selecting this option ensures groups defined in the parent zone have a unique GID across all zones in the Active Directory forest.
   - **RFC 2307 attribute** uses the gidNumber attribute from the RFC 2307 schema to define GID values for the Active Directory groups that you add

to the parent zone. This option requires you to add the RFC 2307 attribute to Active Directory group principals.
- **Use auto-incremented GID** selects the next available GID in the parent zone. In most cases, you should avoid using this option because it does not guarantee unique GIDs.
- **Generate using Apple scheme** generates group GIDs based on the Apple algorithm for generating numeric identifiers from the Active Directory group's objectGuid. This option is only supported for hierarchical zones.

8. Select a method for assigning a new group name to new UNIX group profiles:
   - **SamAccountName attribute** generates the group name for UNIX group profile based on the sAMAccountName value.
   - **CN attribute** uses the common name attribute to define group names for the Active Directory groups you add to the zone. You should only select this option if you verify the common name does not contain spaces or special characters. Otherwise, you should not use this option.
   - **RFC 2307 attribute** uses the cn attribute from the RFC 2307 schema to define group names for the Active Directory groups you add to the zone.
   - **Zone default value** uses the Group name setting from the Group Defaults tab to define group names for the Active Directory groups you add to the zone. In most cases, the default is a variable that uses the sAMAccountName attribute.

     By default, all UNIX group names are lowercase and invalid characters are replaced with underscores.
9. Click **OK** to save your changes.

Add Security Groups to the Parent Zone

The most common way to provision UNIX users is to use a private group identifier as the primary group. With this approach, every user has a unique primary GID that is the same as the UID.

Although not required, another common approach to provisioning UNIX users involves adding a small number of key security groups to the parent zone. For example, if you have a commonly-used group such as All US Employees to which you normally add valid Active Directory users as members, you could add that security group to the parent zone to assign all UNIX users the same primary GID in the parent zone. This approach makes provisioning UNIX users easier because you have already defined Active Directory users as members of that group. If you want to use an Active Directory group to set the primary GID for provisioned users, keep in mind that the size of the group membership can affect the performance of the Zone Provisioning Agent and how long it take to complete provisioning.

If you choose to have the user's primary group defined by Active Directory group membership, the Active Directory group must be in the same Active Directory forest as the users being provisioned. If the Active Directory group is located in another forest, provisioning fails.

If you want to use this approach:

1. Add the security group to the provisioning group for the parent zone (for example, parentZoneName_Zone_Groups).
2. Open the Properties for the parent zone, click the **Provisioning** tab, and define the business rules for the UNIX group profile provisioning associated with the security group.

   At the next update interval, the Zone Provisioning Agent adds a profile for the group to the zone. You can also run the zoneupdate command to add the profile without waiting until the next update interval. For example:

   zoneupdate *zoneName*
3. Click the **User Defaults** tab for the parent zone, select the ellipsis **<...>** option for the Primary Group and select the GID for the group profile that the Zone Provisioning Agent added to the zone.

Defining The Business Rules For New Users In The Parent Zone

In addition to the business rules for group profiles, you configure similar rules for new UNIX user profiles. When you use hierarchical zones, you typically want to configure these business rules for the parent zone so that the profile can be inherited in all child zones. Remember that the profile, by itself, does not provide any access to the computers in the child zones, and that you can override any inherited attributes in any zone or on individual computers.

> **Note**: The business rules you define only affect new UNIX user and group profiles. The imported legacy data remains unchanged, and the Zone Provisioning Agent will not modify any attributes on the existing user and group profiles.

To Configure The Business Rules For User Profiles In The Parent Zone

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the top-level parent zone, right-click, then click **Properties**.
4. Click the **Provisioning** tab.

   If you are defining business rules for a parent hierarchical zone and want to establish a "source zone" for profile attributes, click **Advanced**. You can then select the Source zone for any or all user and group profile attributes. If you select Source zone for any attribute on the Advanced Provisioning page, you can click **Browse** to search for and select the zone to use as the source zone. In most cases, selecting a source zone is not necessary if you using hierarchical zones, but this option can be useful if you are migrating from classic to hierarchical zones.
5. Click **Enable auto-provisioning for user profiles**.
6. Click the Find icon to search for and select the "users" zone provisioning group as the Source Group.

   If you followed the recommended naming convention, search for and select parentZoneName_Zone_Users. For example, if the parent zone name is arcadeGlobal, select arcadeGlobal_Zone_Users.

   This is the same group to which you added the Active Directory users associated with imported user profiles as described in Add existing users to the provisioning group for the parent zone.
7. Select a method for assigning a new UID to new UNIX user profiles:
   - **Generate from user SID** generates new UIDs that are guaranteed to be unique in the forest based on the Active Directory security identifier (SID) of the user. Selecting this option ensures users defined in the parent zone have a unique UID across all zones in the Active Directory forest.
   - **RFC 2307 attribute** uses the uidNumber attribute from the RFC 2307 schema to define UID values for the Active Directory users that you add to the zone. This option requires you to add the RFC 2307 attribute to Active Directory user principals. Otherwise, you should not use this option.
   - **Use auto-incremented UID** uses the next available UID in the parent zone. In most cases, you should avoid using this option because it can create UID conflicts with users in other zones.
   - **Use custom ID** enables you to use the employeeId, employeeNumber, or uidNumber attribute as the UID for new users. You should only select the employeeId or employeeNumber attribute if your organization already populates the employeeId or employeeNumber attribute with a unique value for each user account.
   - **Generate using Apple scheme** generates user UIDs based on the Apple algorithm for generating numeric identifiers from the Active Directory user's objectGuid. This option is only supported for hierarchical zones.
8. Select a method for assigning a new UNIX user login name to new UNIX user profiles:
   - **SamAccountName attribute** generates the user login name for new UNIX users based on the sAMAccountName attribute.
   - **CN attribute** uses common name attribute for user names. You should only select this option if you verify the common name does not contain spaces or special characters. Otherwise, you should not use this option.
   - **RFC 2307 attribute** uses the uid attribute from the RFC 2307 schema to define user names for the Active Directory users that you add to the zone. This option requires you to add the RFC 2307 attribute to Active Directory user principals. Otherwise, you should not use this option.
   - **Zone default value** uses the setting from the User Defaults tab for the zone. In most cases, the default is a variable that uses the sAMAccountName attribute.
9. Select a method for assigning a new shell and home directory to new UNIX user profiles.
   - **RFC 2307 attribute** uses the loginShell attribute for the shell and the unixHomeDirectory attribute for home directory from RFC 2307 schema for the default shell and home directory
   - **Zone default value** uses the values you define on the User Defaults tab, which can include runtime variables for the shell and home directory.

     Runtime variables are populated with platform-specific values when a user tries to log on to a UNIX computer. For example, if a user logs on to a Linux computer with a profile that uses the runtime variable for the home directory, the home directory is /home/username. If the user logs on to a Solaris computer, the runtime variable becomes /export/home/username.
10. Select a method for assigning a primary group to new UNIX user profiles.
    - **RFC 2307 attribute** uses the gidNumber attribute from the RFC 2307 schema for primary group values. This option requires you to add the RFC 2307 attribute to Active Directory user principals. Otherwise, you should not use this option.

- **Zone default value** uses the values you define on the User Defaults tab. This setting enables you to use a specific group profile as the primary group for all UNIX users. If you don't change the default value for the primary group on the User Defaults tab, the default primary group is a private group.
- **Private group** uses the user's UID as the primary GID.
- **Active Directory group membership** uses the Active Directory group with the highest priority as the primary UNIX group. With this option, the Zone Provisioning Agent checks which groups a user belongs to and a prioritized list of groups you have defined. If you select this option, click the Configure icon to search for and select the Active Directory groups to include in the prioritized list. This option allows different users to have different primary GIDs in the same zone.
- **Generate using Apple scheme** generates the user's primary group identifier (GID) based on the Apple algorithm for generating numeric identifiers from the Active Directory objectGuid for the user's primary group. Note that the user's primary group must configured for the zone. If the primary group is not configured for the zone, an error will be logged in the Windows Event Log when the user is provisioned. This option is only supported for hierarchical zones.
- **Generate from group SID** generates new primary GIDs based on the user's Active Directory primary group using the Centrify algorithm for generating GIDs.

  If you select the Active Directory group membership option and a user isn't a member of any of the groups in the list of prioritized groups, the Zone Provisioning Agent will not create a UNIX user profile for the user, because it won't be able to determine the primary group. As noted in Add security groups to the parent zone, the most common approach is to have all users assigned the same primary GID in a zone.

11. Click **OK** to save your changes.

    By default, the GECOS field in new UNIX user profiles is populated using the displayName attribute for the user.

How Hierarchical Zones Affect Provisioning

Because hierarchical zones enable profile attributes to be inherited, defining the business rules for new users and groups in the parent zone enables the Zone Provisioning Agent to generate consistent profiles for all child zones.

When you define a UNIX profile for a group or a user in a parent zone, the attributes are automatically inherited by all child zones. For groups, inheritance makes the group GID and group name available in all child zones. For users, inheritance gives every user defined in the parent zone the potential to log on to every UNIX computer. You then use role assignments to control which computers users can actually access, and, once you begin defining custom roles, what they can do on those computers.

By default, all of the attributes in each new profile are inherited from the parent zone. You can then override any of the attributes as needed in each of the child zones or on individual computers on a case-by-case basis. This flexibility enables you to establish a consistent UID and GID namespace across all zones based on unique SID and sAMAccountName values, while granting exceptions to the specific cases where you need them.

For individual computers, UNIX user and group profiles are inherited from the zone the computer has joined. Typically, this is a child zone or the child of a child zone. You can manually override any attribute or set of attributes for individual computers. Any attributes you do not override are inherited from the zone and the business rules you defined for the Zone Provisioning Agent.

Adding New Users to a Provisioning Group and a Role Group

For new Active Directory users to be effective users of a zone, they must be added to the parent zone's "users" provisioning group and to a role group. You can add users to these groups manually using Active Directory Users and Computers or you can update your existing provisioning process for modifying the membership of Active Directory groups to add users to the appropriate groups. The key points to understand are:

- Users are added to a **provisioning group** so that the Zone Provisioning Agent creates a UNIX profile for them. A user must have a complete UNIX profile to be a valid user on UNIX computers. Centrify recommends creating the profile in the parent zone, but you can create the profile in any zone or on individual computers.
- Users are added to a **role group** so that they have a valid role assignment that allows them to log on or perform specific tasks. Initially, you only have two possible role assignments, listed or UNIX Login, but you are likely to create more.

Add The User to a Provisioning Group

Using Active Directory Users and Computers, scripts, or internal procedures, the basic workflow for a new user would be similar to this:

1. A new Active Directory user requests access to UNIX computers.

2. You add the user principal name to an Active Directory group principal. If you are adding the user to the parent zone, you add the user to the "users" provisioning group parentZoneName_Zone_Users.

   If you wanted to create the profile in a child zone instead of the parent zone, you would add the Active Directory user to the childZoneName_Zone_Users. If you use some other naming convention for the provisioning group, you would search for and select that group.

3. The Zone Provisioning Agent monitors this group and at the next interval (or ondemand) creates a UNIX profile for the user in the zone, based on the business rules you defined.

   Note: If you remove a user from the Active Directory provisioning group, the Zone Provisioning Agent removes the UNIX user profile from the zone.

4. You notify the user that a new UNIX profile has been created with information about the login name and initial Active Directory password to use.

Add the User to a Role Group

Users must also have a role assignment for the zone where you want to grant access. A role assignment is required before the UNIX user profile is usable.

Using Active Directory Users and Computers, scripts, or internal procedures, the basic work flow for a new user would be similar to this:

1. A new Active Directory user requests access to UNIX computers.
2. You add the user principal name to the appropriate Active Directory group principal. If you want to allow the user to log on to computers in a child zone, you add the user to the Login role group childZoneName_Role_Login.

   If the user should be recognized but not allowed to log on, you would add the Active Directory user to the childZoneName_Role_Listed. After you have created custom roles, you would search for and select groups based on the specific rights a user needs.

3. Run the Zone Provisioning Agent update command in preview mode to verify your changes. For example:

   zoneupdate /p *zoneName*

4. Check the results of the zoneupdate preview, then run the command without the preview option to execute the business rules for provisioning. For example:

   zoneupdate *zoneName*

Adding a New Unix Group Profile to All Zones

If you want to make a new UNIX group available to all zones, you should first create a new Active Directory group. In most cases, groups are not shared across multiple zones because of the potential for privilege escalation based on group membership. However, the steps for creating a UNIX profile that spans all zones or only the computers in a specific zone are similar.

Using Active Directory Users and Computers, scripts, or your existing provisioning process, the basic workflow for a new group would be similar to this:

1. Create a new Active Directory group for access to UNIX computers in the UNIX Groups organizational unit (ou=UNIX Groups, ou=UNIX).

   For example, if you are creating a new Active Directory group for the denali project team in the parent zone arcadeGlobal, use Active Directory Users and Computers to create a new group named arcadeGlobal_denali.

2. (Optional) Add users to the group if you know who to add.

   For example, if you are creating the group for a new project and you have a list of authorized users for that project, you can click the Members tab to add those Active Directory users to the new group. If those Active Directory users have a valid UNIX profile and role assignment in the zone, their secondary group membership is updated with the new group.

3. Add the new Active Directory group to the appropriate zone provisioning group. If you are adding the group to the parent zone, you add the user to the "groups" provisioning group parentZoneName_Zone_Groups.

   If you wanted to create the profile in a child zone instead of the parent zone, you would add the Active Directory group to the childZoneName_Zone_Groups. If you use some other naming convention for the provisioning group, you would search for and select that group.

4. Run the Zone Provisioning Agent update command in preview mode to verify your changes. For example:

zoneupdate /p *zoneName*

5. Check the results of the zoneupdate preview, then run the command without the preview option to execute the business rules for provisioning. For example:

   zoneupdate *zoneName*

6. The Zone Provisioning Agent creates a UNIX profile for the group in the zone based on the business rules you defined.

   Note: If you remove an Active Directory group from the Active Directory provisioning group, the Zone Provisioning Agent removes the UNIX group profile from the zone.

Using the Zoneupdate Program for Controlled Automation

You can use the zoneupdate.exe program with command line options to provision profiles in controlled way, allowing you to verify that profiles and access rights are defined correctly for subsets of users or groups without affecting the production environment.

At a minimum, you must specify the zone name or canonical name for the zone to use the zoneupdate.exe program. The command line options are similar to the options available on the Provisioning tab when you display a zone's properties.

For example, to use the provisioning properties defined for a zone, you only need to specify the zone name at the command line:

zoneupdate default

If you use the canonical name for the zone, you specify the full path to the zone:

zoneupdate "centrify.com/program data/Centrify/zones/default"

You can override the default provisioning properties for a zone by specifying one or more of the following command line options.

Options are not case-sensitive. If you specify an option more than once, only the last value is used.

/z:ZoneName or /SourceZone:ZoneName	The name of a source zone. If you do not specify a zone name and there's not a source zone defined in the zone's provisioning properties, you cannot use the zoneupdate command to copy user or group attributes from one zone to another. A source zone is required for classic zones. It is optional for parent hierarchical zones, but can be useful if you are migrating from classic to hierarchical zones.
/d:DomainName or /Domain:DomainName	The name of the domain to process. If you do not specify a domain name, the zoneupdate program processes the Active Directory domain to which the computer is joined.
/dc:DCName or /DomainController:DCName	The name of the target domain controller to connect. No option - This will use the default domain controller of target domain.
/uu:Option or /UserUid:Option	The method to use to set the user's numeric identifier (UID) value. You can specify any one of the following values: Auto to generate UIDs based on the Active Directory domain name and the RID of a user object. This setting is equivalent to selecting Generate from user SID in the Provisioning tab. AppleScheme to generate UIDs based on the Apple algorithm for generating numeric identifiers from the Active Directory user object's objectGuid. This setting is equivalent to selecting Generate using Apple scheme in the Provisioning tab. RFC2307 to use the uidNumber attribute in the Active Directory RFC2307 schema for the user's UID value. ZoneDefault to use the UID defined on the User Defaults tab for the zone. If there's no default value, the Next UID value for the zone is used. SourceZone to copy the UID from the same user in a specified source zone. EmployeeId to copy the UID from the employeeId attribute of the user object. EmployeeNumber to copy the UID from the employeeNumber attribute of the user object. uidNumber to copy the UID from the uidNumber attribute of the user object. If you don't use one of these values, you can set the UID to not have any value. For example: /uu:empty If you use this setting, users will have an incomplete profile in the zone.
/un:Option or /UserName:Option	The method to use to set the user's name. You can specify any one of the following values: sAMAccountName to use the Active Directory user's sAMAccountName attribute as the UNIX user name. CN to use the user's common name (CN) attribute as the UNIX user name. RFC2307 to use the uid attribute in the Active Directory RFC2307 schema as the UNIX user name. ZoneDefault to use the user name defined on the User Defaults tab for the zone. If there's no default value zone, the

	sAMAccountName is used. SourceZone to copy the user name from the same user in a specified source zone. If you don't use one of these values, you can set the user name to an explicit value. For example: /un:hunter
/us:Option or /UserShell:Option	The method to use to specify the user's default login shell. You can specify any one of the following values: RFC2307 to use the loginShell attribute in the Active Directory RFC2307 schema as the default shell. ZoneDefault to use the shell specified on the User Defaults tab for the zone. SourceZone to copy the shell defined for the user in a specified source zone. If you don't use one of these values, you can set the login shell using an explicit value. For example: /us:/bin/bash
/uh:Option or /UserHomeDirectory:Option	The method to use to specify the user's default home directory. You can specify any one of the following values: RFC2307 to use the unixHomeDirectory attribute in the Active Directory RFC2307 schema for a user as the default home directory. ZoneDefault to use the home directory specified on the User Defaults tab for the zone. SourceZone to copy the home directory defined for the user in a specified source zone. If you don't use one of these values, you can set the home directory to an explicit value. For example: /uh:/home/hunter
/ug:Option or /UserPrimaryGroup:Option	The method to use to specify the user's primary group identifier. You can specify any one of the following values: AppleScheme to generate the user's primary group identifier (GID) based on the Apple algorithm for generating numeric identifiers from the Active Directory objectGuid for the user's primary group. Note that the user's primary group must configured for the zone. If the primary group is not configured for the zone, an error will be logged in the Windows Event Log when the user is provisioned. This setting is equivalent to selecting **Generate using Apple scheme** in the Provisioning tab. PrimaryGroupSID to generate the user's primary group identifier (GID) based on the Centrify algorithm for generating numeric identifiers from the Active Directory security identifier of the user's primary group. PrivateGroup to set the user's primary GID value to be the same as the user's UID value. RFC2307 to use the gidNumber attribute f in the Active Directory RFC2307 schema as the primary group identifier or a user. ZoneDefault to use the primary group specified on the User Defaults tab in the zone. If there's no default value, zoneupdate.exe will stop provisioning the user. SourceZone to copy the primary group defined for the user in a specified source zone. example: /ug:empty If you use this setting, users will have an incomplete profile in the zone. GroupMembership to set the user's primary GID based on the user's Active Directory group membership. If a user is a member of the Active Directory groups ops-mgrs and ops-labs and one of those groups has a UNIX profile in the zone but not the other, the group with the UNIX profile in the zone will be used as the primary GID for the user. If both group have a UNIX profile in the zone, the one with higher priority will be used. You can set the priority for selecting the primary group to use in the Access Manager console. If the priority is the same, zoneupdate.exe will stop provisioning the user. If you don't use one of these values, you can set the primary GID to not have any value.
/uc:Option or /UserGecos:Option	The method to use to specify the user's GECOS field. You can specify any one of the following values: RFC2307 to use the gecos attribute in the Active Directory RFC2307 schema for a user. ZoneDefault to use the value defined for the GECOS field on the User Defaults tab for the zone. If you don't use one of these values, you can set the primary GID value to an explicit value. For example: /uc:Thompson, Hunter S.
/gg:Option or /GroupGid:Option	The method to use to set the group numeric identifier (GID) value. You can specify any one of the following values: Auto to generate the GID based on the Active Directory domain name and the RID of a group object. This setting is equivalent to selecting Generate from group SID in the Provisioning tab. AppleScheme to generate GIDs based on the Apple algorithm for generating numeric identifiers from the Active Directory group object's objectGuid. This setting is equivalent to selecting Generate using Apple scheme in the Provisioning tab. RFC2307 to use the gidNumber attribute in the Active Directory RFC2307 schema for the group GID value. ZoneDefault to use the GID defined on the Group Defaults tab for the zone. If there's no default value, the Next GID value for the zone is used. SourceZone to use the GID defined for the group in a specified source zone. If you don't use one of these values, you can set the GID to not have any value. For example: /gg:empty If you use this setting, groups will have an incomplete profile in the zone.
/gn:Option or /GroupName:Option	The method to use to set the group name. You can specify any one of the following values: samAccountName to use the Active Directory group samAccountName attribute as the UNIX group name. CN to use the group's common name (CN) attribute as the UNIX group name. RFC2307 to use the cn attribute in the Active Directory RFC2307 schema as the UNIX group name. ZoneDefault to use the group name defined on the Group Defaults tab for the zone. If there's no default value, the sAMAccountName is used. SourceZone to copy the group name from the group in a specified source zone. If you don't use one of these values, you can set the group name to an explicit value. For example: /gn:apps-lab
	An Active Directory group to use to populate a Centrify zone with users. Use the sAMAccountName and, optionally, the domain name to identify the group. For example, to use the Active Directory engineers group in the currently connected

/u:ADGroupName or /UserSource:ADGroupName	domain to populate users in the default zone: zoneupdate /u:engineers default To use the Active Directory engineers group in a specific domain, you can use the /d:DomainName option or group_name@domain_name. For example to use the Active Directory engineers group in the testdomain.org domain to populate users in the default zone: zoneupdate /u:engineers@testdomain.org default
/g:ADGroupName or /GroupSource:ADGroupName	An Active Directory group to use to populate a Centrify zone with groups. Use the sAMAccountName and, optionally, the domain name to identify the group. For example, to use the Active Directory employees group in the currently connected domain to populate groups in the default zone: zoneupdate /g:employees default
/v or /Verbose	Display detailed information about the provisioning of users and groups. When you use this option, the output format is: Group: groupname:gid User: uid:username:shell:home:primarygid
/p or /Preview	Preview the users or groups to be provisioned or removed. In preview mode, the zoneupdate.exe program does not create or remove any UNIX profiles.
/el or /EventLog: Level	Enable logging to the Event log. You can use the Event Viewer to check the log results. For the log level, you can specify any one of the following values: None - don't write any provisioning activities to the Event log. This is the default setting. Normal - Write only the name of the provisioned users and groups to the Event log. Verbose - Write the UNIX profiles for the provisioned users and groups to the Vent log.
/l or /Log:Level	Enable logging and set the level of detail recorded in the log file. For the log level, you can specify any one of the following values: Error to log only error messages. Warning to log warnings and error messages. Information to log informational messages, warnings, and errors. Verbose to log all messages, including details about the user and group profiles created or removed. Logging is off by default. If you enable logging, the default file location for the log file is: C:\Users\user_name\AppData\Roaming\Centrify\Zone Provisioning Agent\Log You can change the default log file path by modifying the following registry key: HKEY_LOCAL_MACHINE\Software\Centrify ZPA\LogLevel

Using Any Active Directory Attribute in a Profile

In addition to the provisioning properties you can set for a zone using Access Manager, you can manually configure the Zone Provisioning Agent to use any attribute in Active Directory to define a value for any field in automatically-provisioned UNIX user or group profiles. For example, if your organization uses a custom attribute, such as org_global_id, for all users, you can manually configure the Zone Provisioning Agent to use that attribute for the numeric user identifier (UID) in automatically-generated user profiles.

To manually specify an Active Directory attribute to use in a UNIX profile:

1. Open Microsoft ADSI Edit.
2. Select a target zone, right-click, then click **Properties**.
3. Select the **description** attribute, then click **Edit**.
4. Type a profile provisioning attribute and specify the Active Directory attribute to use for the profile.

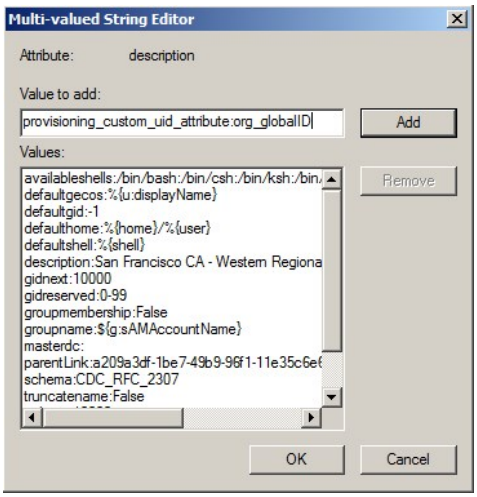
   The valid provisioning attributes are:

   provisioning_custom_uid_attribute
   provisioning_custom_gid_attribute
   provisioning_custom_primary_group_attribute
   provisioning_custom_user_unixname_attribute
   provisioning_custom_group_unixname_attribute
   provisioning_custom_home_directory_attribute
   provisioning_custom_shell_attribute

   The format for the entry is:

   provisioning_custom_uid_attribute:attribute_name

Replace *attribute_name* with the Active Directory attribute you want to use. For example:



5. Click **Add**, then click **OK**.
6. Run the Zone Provisioning Agent update command in preview mode to verify your settings. For example:

   zoneupdate /p *zoneName*

7. Check the results of the zoneupdate preview, then run the command without the preview option to execute the business rules for provisioning. For example:

   zoneupdate *zoneName*

If the Active Directory attribute type is different from the target profile value, the Zone Provisioning Agent attempts to convert the data type. If the data conversion fails, the Zone Provisioning Agent reports an error and stops the provisioning process.

Provisioning Users When Across Trusted Domains

The Zone Provisioning Agent includes two utilities in its Tools folder to assist you in provisioning users when there are trust relationships between domains. These two utilities are CopyGroup and CopyGroupNested. These utilities enable you to mirror group membership or a group hierarchy from a trusted domain and forest in a target domain and forest.

To use these command line utilities, you must have an account that can log on to the trusted source domain and the target domain. The account should also have read permission on the source domain and permission to update the target domain.

For example, assume you have configured the AJAX domain to have a one-way trust with the ACME domain and you have your Active Directory users and groups defined in the ACME domain. If you want to allow the users and groups in the ACME domain to log on to computers that are joined to the AJAX domain, you can log on to the AJAX domain controller with an account that has administrative privileges in both the AJAX and ACME domains, then run the CopyGroup utility to mirror the group membership from a group in the ACME source domain as zone users in the AJAX target domain.

For more information about the command line arguments and options for these utilities, see the usage message displayed for each utility.

Monitoring Provisioning Events

The Zone Provisioning Agent writes events to the Windows event log. You can use tools that monitor the event log to notify you of specific provisioning events. The following table lists the event identifiers and messages the Zone Provisioning Agent records.

1	Information	Summarizes the result after a provisioning run. Centrify zones updated. Domain controller: domain_controller_name Container: container_name Start time: start_time End time: end_time Successful: successful_message Failure: failure_message
2	Error	Indicates that provisioning failed for a specific domain because the domain was not found. Domain domain_name not found.

3	Error	Indicates that provisioning failed because the domain controller was not accessible. Domain server domain_controller_name is down.
4	Error	Indicates that provisioning failed because the domain was not operational. Domain domain_name is not operational.
5	Error	Indicates that provisioning failed without a specific cause. Failed to update zones in domain domain_name on domain controller domain_controller_name.
6	Error	Indicates that provisioning failed because there was a problem with the agent connecting to the Active Directory. Failed to update zones in domain domain_name on domain controller domain_controller_name. Error on Active Directory service.
7	Error	Indicates that provisioning failed because there was an unexpected error when updating the zones in a specific domain. Unexpected error when updating the zones in domain domain_name on domain controller domain_controller. Details: detail_message.
8	Information	Provides verbose user provisioning information, including details about the provisioned user profile. User provisioned to a Centrify zone. User: user Zone: zone UID: uid Name: name Shell: shell Home directory: home_directory Primary group: primary_group Gecos: gecos
9	Information	Provides basic user provisioning information. User provisioned to a Centrify zone. User: user Zone: zone
10	Information	Provides verbose user de-provisioning information, including details about the user profile removed. User removed from a Centrify zone. User: user Zone: zone UID: uid Name: name Shell: shell Home directory: home_directory Primary group: primary_group Gecos: gecos
11	Information	Provides basic user de-provisioning information. User removed from a Centrify zone. User: user Zone: zone
12	Information	Provides verbose group provisioning information, including details about the provisioned group profile. Group provisioned to a Centrify zone. Group: group Zone: zone GID: gid Name: name
13	Information	Provides basic group provisioning information. Group provisioned to a Centrify zone. Group: group Zone: zone
14	Information	Provides verbose group de-provisioning information, including details about the group profile removed. Group removed from a Centrify zone. Group: group Zone: zone GID: gid Name: name
15	Information	Provides basic group de-provisioning information. Group removed from a Centrify zone. Group: group Zone: zone
16	Warning	Indicates that the Zone Provisioning Agent received a warning message during the provisioning process. Warning occurred when provisioning a Centrify zone. Zone: zone Details: detail_message
17	Error	Indicates that provisioning failed because there was a problem with the permissions on the account used to run the Zone Provisioning Agent. Insufficient permission to setup the log file. Please contact your local administrator. Details: detail_message
18	Error	Indicates that provisioning failed because there was a problem with creating the log file in the log file location. Failed to create the log file. Please contact your local administrator. Details: detail_message
19	Information	Indicates that provisioning is paused to allow another provisioning cycle to complete. Zone Provisioning Agent failed to start another provisioning cycle at current_time because the previous provisioning cycle is not yet complete. The provisioning request is pending until Zone Provisioning Agent finishes the previous provisioning cycle.
20	Error	Indicates that provisioning failed because the computer was not found in the domain being provisioned. This computer is not joined to a domain or the domain is not available.
21	Error	Indicates that provisioning failed because there was a problem loading domain information. Failed to load domain information. No zone will be provisioned. Error: error_message

22	Error	Indicates that provisioning failed because there was a problem with the functional operation of the domain. Functional error occurred with domain domain_name.
23	Error	Indicates that provisioning failed because authentication failed for the account used to run the Zone Provisioning Agent. Domain domain_name authentication failed.
24	Error	Indicates that provisioning failed because the authentication system failed. Domain domain_name authentication (system) failed.
25	Error	Indicates that provisioning failed because there was an unexpected error when loading a specific domain. Unexpected error occurred when loading domain domain_name.
26	Error	Indicates that provisioning failed because the Active Directory forest was not found. Forest forest_name not found.
27	Error	Indicates that provisioning failed because the root domain controller was not accessible. Forest server server_name is down.
28	Error	Indicates that provisioning failed because the forest was not operational. Forest forest_name is not operational.
29	Error	Indicates that provisioning failed because there was a problem with the functional operation of the forest. Functional error occurred with forest forest_name.
30	Error	Indicates that provisioning failed because authentication failed at the forest level for the account used to run the Zone Provisioning Agent. Forest forest_name authentication failed.
31	Error	Indicates that provisioning failed because the authentication system failed at the forest level. Forest forest_name authentication (system) failed.
32	Error	Indicates that provisioning failed because there was an unexpected error at the forest level. Unexpected error occurred when loading forest forest_name.
33	Error	Indicates that provisioning failed because there was an error during the provisioning process. Error occurred when provisioning a Centrify zone. Zone: zone Details: detail_message
34	Warning	Indicates that provisioning might be incomplete because the account used to run the Zone Provisioning Agent does not have permission update zone in a specified domain. Insufficient permission to update the zones in domain domain.

Adding New Profiles Manually

Provisioning groups enable automated provisioning of UNIX profiles for users and groups with the Zone Provisioning Agent. Server Suite does not require you to use provisioning groups and business rules to create new users and groups. You can also manually create UNIX users and groups in any zone using the Access Manager console, ADedit, or custom scripts. Adding profiles manually to a zone provides you with control over the definition of individual UNIX profiles on a zone-by-zone basis.

Validating Operations After Deploying

This section provides sample activities for creating test cases and performing a formal validation of the Server Suite deployment. Although not required, executing a set of test cases that exercise Server Suite functionality will help you validate operations before extending the deployment to additional computers in the enterprise.

The specific use case scenarios and test cases you execute will depend on your organization's goals and requirements.

Understanding Testing as Part of Deployment

Most organizations initially deploy in a lab environment that simulates the production environment. The lab environment allows you to test the planned changes to system and user management processes. For example, if you plan to automate migration using scripts, you should build and test the operation of your tools to verify they work as you intend. After testing in a lab, most organizations move to a pilot deployment with a limited number of computers and users to continue verifying that authentication, authorization, and directory services are all handled properly in a real-world environment before moving to a full-scale migration across the enterprise.

In many cases, the pilot deployment also requires more formal testing of specific use cases to validate the deployment before rolling out software to additional computers. This phase of the deployment may also include activities designed to help users transition to Active Directory.

To validate the deployment:

- Execute test cases that verify authentication.
- Execute test cases the verify authorization rules for login access and restricted access.
- Execute test cases that verify the provisioning process.
- Execute test cases that address issues unique to your organization or your user community.

Other activities you may want to perform as part of the validation process include:

- Test configuration changes and customize configuration parameters.
- Document test results.
- Troubleshoot any authentication or authorization issues, if any are found.
- Train staff on new procedures.
- Communicate process changes to the users who are migrating to Active Directory.

  For example, if you plan to eliminate local account access, implement stricter password policies, or apply new access controls, you should prepare the user community for these changes.

Validating Basic Authentication and Password Policy Operations

Before you begin testing organization-specific scenarios, such as the integration of migration scripts or customized access control policies, you should verify basic operations are handled as expected. At a minimum, you should perform some of the following tests to verify basic operations:

- Verify a UNIX profile exists for the Active Directory users and groups to be used in testing.
- Verify the UNIX computer has successfully joined an Active Directory domain and the computer account has been created correctly.
- Verify an Active Directory user assigned the UNIX Login role is authenticated and can access computers in the zone used for testing.
- Verify migrated users assigned the UNIX Login role can log on using their UNIX or Active Directory user name and Active Directory password.
- Verify an Active Directory user assigned the Listed role has a valid UNIX profile, but cannot log on to computers in the zone used for testing.
- Verify that workstation authorization or account lockout policies defined in Active Directory are enforced.
- Verify that password management policies defined in Active Directory are properly enforced.
- Verify that a previously authenticated user is authenticated successfully when the UNIX lab computer is offline.
- Verify that common lookup commands and commands that require user and group information work as expected.

You can verify authentication, authorization, and password policy operations by setting options in Active Directory and attempting to log on and log off the computers in the pilot deployment.

Running Commands on the Unix Computer to Verify Operations

To confirm that a UNIX computer is a member of the Active Directory domain, you can run commands that retrieve information from Active Directory on the

UNIX lab computer itself or by viewing the UNIX computer's account information in Active Directory Users and Computers or the Access Manager console.

Verify the Computer Is Joined to Active Directory

To verify a computer is joined to the Active Directory domain and is retrieving information from Active Directory by running commands on the UNIX computer:

1. Log on to the UNIX computer.
2. Type the following command to retrieve information about the computer's connection to Active Directory:

   adinfo

   This command returns basic information such as the host name for the computer, whether the computer is joined to the domain, and whether the computer is currently connected to Active Directory. For example:

   Local host name: magnolia
   Joined to domain: ajax.org
   Joined as: magnolia.ajax.org
   Current DC: ginger.ajax.org
   Preferred site: Default-First-Site-Name
   Zone: ajax.org/Centrify/Zones/default <!--TODO: company name in file> Last password set: 2017-12-21 11:37:22 PST
   CentrifyDC mode: connected

   For more detailed information about the environment, you can use --diag or other options with the command. For information about the options available and the information displayed for each option, see the adinfo man page.
3. Type the following command to verify that the adclient process is running:

   ps -aeflgrep adclient

   The command should return output similar to the following:

   root 1585 1 0 14:50 ? 00:00:29 adclient
4. Type the following command to confirm that lookup requests use the information in Active Directory:

   getent passwd

   The command should list all of the Active Directory user accounts that are members of the zone and all local user accounts in the /etc/passwd file format. For example:

   ben:x:601:100:Ben Waters:/home/ben:/bin/bash
   ashish:x:501:100:Ashish Menendez:/home/ashish:/bin/bash
   sunni:x:900:100:Sunni Ashton:/home/sunni:/bin/bash
   jolie:x:502:100:Jolie Ames:/home/jolie:/bin/bash
   pierre:x:1001:100:Pierre Leroy:/home/pierre:/bin/bash
5. Review the contents of the /var/log/messages file and look for messages that indicate authentication problems or failures.

Verify Authentication for an Authorized User

To verify that an authorized Active Directory user can log on to a UNIX computer:

1. Restart the computer to display a logon screen or prompt.
2. Log on with the Active Directory user account you created for testing and provide the Active Directory password for that account.
   - The user account must have a complete UNIX profile.
   - The user account must be assigned the UNIX Login role.
   - If the user account has been configured to set a new password at the next logon, you should be prompted to change the password. In this case, you must type and confirm the new password before continuing.
3. Log on using the UNIX login name for the user account and the Active Directory password.
4. Type commands to check the UID and GID assignments, home directory ownership, and other information for the logged on user.

Test Additional Administrative Tasks

You may want to try other typical administrative tasks that you expect to perform in the production environment. For example, you may want to test and verify the following tasks:

- Changing the password for an Active Directory user using the passwd or adpasswd command on a UNIX computer changes the user's Active Directory password for Windows computers.
- Logging on as a user from another trusted Active Directory domain or another trusted forest is successful when you specify the user's fully-qualified domain name (for example, milo.cutter@paris.arcade.com).
- Setting a user's effective group membership using the adsetgroups command.
- Logging on using previously cached credentials. Offline authentication enables users to log on when computers are disconnected from the network or have only periodic access to the Active Directory domain. For example, users who have laptop computers must be able to log on and be successfully authenticated when they are not connected to the network.

Resolving Issues in the Pilot Deployment

Executing a formal test plan is intended to help you uncover issues that need to be resolved, troubleshoot any unexpected behavior, and correct any potential problems before end-users are affected. The pilot deployment enables you to deploy Server Suite software packages on a subset of typical users in the production environment in a controlled way. You can then use the pilot deployment to evaluate server and network load and how adding new computers and users to the Active Directory affects your environment. The initial deployment also allows you to closely monitor the experience of the user community participating in the pilot program.

With the pilot deployment, you can also develop and refine your processes and operational expertise before you roll out Server Suite authentication and authorization services to the entire organization.

You can install Server Suite Agents at any time without affecting any user or computer operations. Installing the Server Suite Agent on UNIX computers has no effect until you join the computer to the domain. Setting up the initial zone or set of zones does not affect the operation of any existing Active Directory infrastructure or Windows environment. Therefore, you can install the software for the pilot whenever it is convenient to do so.

Before you join computers to the domain, you must define and assign appropriate roles to the user community. Users who don't have role assignments will not be allowed to log on to any computers. It is essential for you to test and validate role definitions and assignments to ensure users won't be locked out of the computers they need access to when computers join the domain.

Preparing Training and Documentation for Administrators and Users

The deployment team should develop and deliver training for end-users, technical support personnel, help desk operators, and account fulfillment personnel. This role-based internal training will help new team members come up to speed and also help with the resolution of technical issues.

You should also train staff members to understand that there will be two fulfillment processes in place during migration: the legacy account fulfillment process for computers that have not joined the domain and a new account fulfillment process for computers that have joined an Active Directory domain. Both fulfillment processes should be clearly documented and staff should be trained on how to determine which process to use. For example, training material should indicate how the UNIX provisioning team can determine whether a computer is in a zone, so that members know whether to use the legacy process or the new process.

After a computer is migrated to Active Directory, you should not allow any local account provisioning on that computer. You should also be sure that this is clearly documented in training materials, especially if you don't have centralized management of account creation policies. If you don't prevent local account provisioning, orphaned and noncompliant UNIX accounts can continue to exist, may create conflicts in the UID and GID namespace, and create audit compliance issues because they are not included in required reports.

As you migrate each set of computers to an appropriate zone, you should also notify all affected users before you complete the migration. This notification can take the form of an email, voicemail, meeting with project personnel or management, or any other logical combination. Notifying users in advance helps to reduce the number of account lockouts caused by UNIX users attempting to log on using their old UNIX password on migrated computers.

Deploying to the Production Environment

After the initial deployment is stable and you have migrated existing users and groups successfully, you can begin moving the rest of your UNIX computers, users, and groups to Active Directory. In most cases, this migration is done in stages by repeating the tasks described in this guide for additional target sets of computers, users, and groups. After each stage, you should allow a period of time for monitoring and resolving issues for the migrated user population. Your deployment plan should include a schedule for when different sets of users are to be migrated and an analysis of how those users should be placed into zones according to your migration plan.

In general, you should migrate an increasing number of computers into zones in each stage of the production deployment. For example, in the first round of

migration, you might migrate 15% of the computers into the first set of zones. You should then allow time in the schedule to troubleshoot and resolve issues to ensure that the migration was successful. In the next phase, you then might migrate another 25% of the computers. After you determine the second phase of the migration is successful, you might migrate the remaining computers into the remaining zones.

> **Note**: Whenever possible, you should also plan to migrate all of the computers that have been identified for a particular zone at the same time. Migrating all of the computers in a zone at the same time helps to reduce user confusion over which password to use when authenticating.

Training the Support Staff for a Production Deployment

You should provide the following information or training to the IT support staff who are responsible for a set of users to be migrated to Active Directory:

- **Review of the deployment project plan**. Have the support staff read the deployment plan and review any changes to policies or procedures that deployment will entail.
- **Schedule for deployment**. Make sure members of the support staff are aware of when the deployment is scheduled to take place and that they will be available at that time and for a reasonable period thereafter.
- **Location of documentation**. Make sure all internal, operating system, and Server Suite-specific documentation is available so that support staff can use those documents to help them resolve any end-user issues that arise during the production deployment.
- **Pilot deployment experience and feedback**. Explain the result of the pilot deployment, including any issues encountered during the pilot and the resolution for each issue.
- **Common Windows and Active Directory administrative tasks**. For support staff members who are familiar only with supporting UNIX-based computers, provide training about Windows and Active Directory concepts and administration, as appropriate. If administrators will be using Windows-based programs or scripts to manage UNIX users and computers, they may need training specific to those tools. For example, administrators may need training to use Active Directory Users and Computers, Visual Basic scripts, or Access Manager console to manage Active Directory data.
- **Common UNIX administrative tasks**. For support staff members who are familiar only with supporting Windows-based computers, provide training about UNIX concepts and administration, as appropriate. If administrators will be using UNIX-based programs or scripts to manage UNIX users and computers, they may need training specific to those tools. For example, administrators may need training to create and use ADedit or LDAP scripts to manage Active Directory data.
- **Common access control and privilege management tasks**. Make sure members of the support staff are familiar with the tasks described in the *Administrator's Guide for Linux and UNIX*, and provide hands-on training in performing the most common of those tasks.
- **Internal policies and procedures specific to your network and business environment**. Create an operations handbook with details about common scenarios the support staff may be required to address, such as adding new UNIX computers or users to Active Directory.
- **Reporting and tracking issues related to Server Suite software**. Make sure support staff members know how to report issues or problems with authentication, authorization, or directory services. If your organization uses a bug or problem-ticket system for tracking issues, set up a new subject area for Server Suite-related issues.

Preparing the User Community in a Production Deployment

As you prepare to migrate a set of users to Active Directory, you should provide training or informational materials to inform that user community about what to expect. For example, if your organization has decided to implement policies that prevent locally-defined user accounts from accessing some computers, be sure that the user community affected by this policy understands the change. Similarly, if your organization has decided to eliminate service accounts or restrict access to computers previously available, you should communicate these changes and notify users about any migration issues that may affect file access permissions and file ownership.

When you are ready to migrate a specific set of users, you should inform the user population about the upcoming deployment by providing the following information:

- **Schedule for deployment**. Make sure that department managers and end-users know when the switch to Active Directory is scheduled to occur.
- **Computers and applications affected**. Make sure that department managers and end-users know if their workstations or the servers they access for business applications are included in the deployment. If users need access to a computer that is being added to an Active Directory domain, they need to know whether their user account is in the same domain as the computer or a different domain. If there are applications hosted on a computer that is being added to an Active Directory domain, users need to know how this will affect access to the hosted application. For example, users may need to select a domain when logging on, or log on using the user_name@domain_name format.
- **Active Directory account information**. Make sure that end-users know their Active Directory account information and understand that they must use their Active Directory password to access their UNIX workstations after the deployment is complete. You should inform users about the valid logon names and formats they can use, the Active Directory password assigned to their account if it is a new account, whether they are required to change their password when they next log on, and any password complexity rules you have implemented. Active Directory may lock accounts if users attempt to log on using their UNIX password, which could result in a large number of Help Desk requests for password resets.
- **Changes to access policies**. Make sure that department managers and end-users are aware of any changes to access control policies. For example, if you are using group policies to deny access to some users or groups who could previously log on to a computer, you should inform those

users or groups of the change and that it will take effect after the migration to Active Directory.

Populating Zones in a Production Environment

In planning your deployment, you should have determined your basic zone requirements and how you will migrate existing user communities to Active Directory. Based on your analysis, you should have a zone design with one or more parent zones and the child zones for each parent to define a candidate set of users and groups with the potential to access a given set of computers.

Typically, you should focus on one zone at a time, importing and mapping the existing users to Active Directory accounts. You should also determine whether you need to create new Active Directory accounts for any of the existing users or groups you are importing. If possible, you should use Active Directory group membership and role assignments to manage access for UNIX users and groups, you import.

Joining a Domain in a Production Environment

In smaller organizations or organizations where individual users have permission to join their own workstation to the Active Directory domain, you can run the adjoin command interactively on individual computers. This option works well when computers are distributed across many different domains or when individual users are joining their own workstation to the domain.

In larger organizations, however, you may want to use a custom script to remotely join a group of UNIX computers to an Active Directory domain. If you develop a custom script for joining a domain, the script should restart services or reboot the computers where it runs.

After joining a domain, you should monitor computers closely for a few days before extending the deployment to additional computers.

If the join operation fails or users cannot log on, you can run the adleave command to restore the computer to its previous state.

Defining Role-Based Access for Users and Computers

By default, Server Suite includes two roles—the listed role and the UNIX Login role—that are required for migration. This section discusses additional role-based controls you can define for better management of privileged access and authorized activity.

> **Note**: If you have well-defined access rules and command privileges in sudoers configuration files, you can import those definitions and use them as the basis for creating custom roles in Access Manager. For information about importing sudoers files and converting the imported definitions into roles, see the *Administrator's Guide for Linux and UNIX*.

Addressing the Business Problem of Role-based Security

Privilege management and role-based access controls are approaches to the basic business problem of securing an enterprise's key computer systems and sensitive data. Restricting access based on a user's role or specific job requirements can require you to make some difficult decisions about who has access to what and why access is granted or denied. These decisions also have the potential to disrupt user activity or existing business processes. Therefore, you should do thorough planning to identify the roles to implement, who should have permission to execute privileged commands, and who should have restricted access.

Defining the appropriate rights for users in different roles often requires negotiation with different groups in the organization to achieve the right balance of security and functional capability. Before implementing a solution, you should have these conversations and set expectations about what will change in the user's environment.

Creating a Root-Equivalent Role Definition

One of the first roles you should plan to create is an administrative role that is equivalent to specifying ALL:ALL in a sudoers file or giving users access to the root password on their computers. The purpose of this role definition is to allow selected users to execute privileged commands on a regular basis. The role definition allows them to execute commands without being given the root password or having privileges hard-coded in individual sudoers files on multiple computers.

Because this role definition enables system administrators to execute privileged commands without the root password, you can improve security for the organization and reduce the chance of an audit finding for access to the root password.

You can create this role definition in a parent zone or a child zone to control its scope. In most cases, you should only assign the role in a child zone or on an individual computers.

Define the Right for Running All Commands

Rights and roles are defined at the zone level and inherited down the zone hierarchy. If you define a right in the top-level zone, it is available in all child zones. If you define a right in a child zone, it can be used in that zone and any of its child zones. Similarly, you can define roles in the top-level parent or any child zone, depending on where you want to make the role available. In this example, the right to run all commands as the root user is defined in a top-level parent zone.

The following instructions illustrate how to define a right for running all commands using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Server Suite Windows API are available in other guides, the *Server Suite Software Developer's Kit,* or in community forums on the Delinea website.

To define a right for running all commands as root:

1. Open Access Manager.
2. Expand Zones and select the top-level parent zone.
3. Expand **Authorization > UNIX Right Definitions**.
4. Select **Commands**, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a description for this right, then define the right to run all commands like this:
    - Type an asterisk (*) in the Command field to indicate all commands are allowed.
    - Select **Specific path** and type an asterisk (*) in the field to indicate that any path is allowed.
6. Click the Restricted Shell tab and deselect the **Can be used in a restricted role** option if you want to prevent this command from being used in a role that uses a restricted shell environment.

7. Click the Run As tab to verify the command can be used by dzdo and is set to run as root by default.
8. Click **OK** to use the default environment variable settings and command attributes.

   Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Create a Role Definition for Running All Commands

After you have defined the right to allow a user to run any command with root privileges, you can create a role definition for that right. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

To create a role definition with the right to run all commands as root:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the role definition.
3. Expand Authorization.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a name and description for the new role, then click **OK**.

   For example, type a name such as root_equivalent and descriptive text such as Users with this role can run any command with root privileges.

   Optionally, you can select **Allow local accounts to be assigned to this role** if you want to assign both Active Directory users and local users to the role. This option is only available when you first create a role definition. You can also click **Available Times** if you want to limit when the role is available for use. By default, roles are available at all times.

   If you are using the UNIX Login role to grant access to computers in the zone and want to use the default auditing level of **Audit if possible**, you can click **OK** then skip to Step 8.
6. If you are not assigning the UNIX Login role to grant access to computers, click the System Rights tab and select the following options:
   - Password login and non-password (SSO) login are allowed
   - Non-password (SSO) login is allowed
   - Login with non-Restricted Shell

     Note that you cannot set these system rights if you selected the option to allow local users to be assigned to this role.
7. If you don't want to use the default auditing level, click the Audit tab.
   - Select **Audit not requested/required** if you have the auditing service enabled but don't want to audit user activity when this role is used.
   - Select **Audit if possible** to audit user activity where you have the auditing service enabled.
   - Select **Audit required** to always audit user activity. If the auditing service is not installed or not available, users in this role are not allowed to log on.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select the right you defined for running all commands as root, then click **OK**.

Assign an Active Directory Group to the Role

As discussed in previous chapters, you should associate Centrify role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership. If you are using the recommended deployment structure and naming conventions, you would create a new Active Directory group in the ou=User Roles, ou=Centrify organizational unit using the format ZoneName_Role_RoleName. For example, you would create an Active Directory group named sanfrancisco_role_rootequivalent. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.

2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select **Role Assignments**, right-click, then click **Assign Role**.
5. Select the role definition you created for root-level access, such as root_equivalent, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role.
    - Select **Group** as the object to find.
    - Optionally, type all or part of the group name.
    - Click **Find Now**.

      Select the group you created for the role in the results, then click **OK**.
7. Click **OK** to complete the assignment.
8. Add members to the Active Directory security group for the role definition using Active Directory Users and Computers, an internal script, or another tool.
9. Test the role assignment by checking whether the user you added to the Active Directory group can execute privileged commands using dzdo in place of sudo.

   dzdo /usr/share/centrifydc/din/adflush

   Details about commands that are executed with dzdo are logged to the secure syslog facility on the computer where they were executed.

Creating a Restricted Role for a Shared Service Account

The root-equivalent role definition provides centralized management for a limited number of administrators who have permission to execute all commands on selected computers. Another common reason for defining a role is to execute privileged commands associated with a service account. In many organizations, service account passwords are known by multiple users, making them a security risk. For example, all of the database administrators in the organization might know the password for an oracle service account, an account with permission to perform privileged database operations. Because the password is shared information, it presents a security risk and a potential audit finding that might have costly consequences.

Setting up a role definition for a service account involves creating a command right for switching to the service account user and defining a PAM access right for role.

Define the Right for Switching to a Service Account

The steps for defining a right for switching to the service account user are similar to defining the rights for the root-equivalent user, but the definition is more restrictive.

To define a right for switching to a service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand **Authorization > UNIX Right Definitions**.
4. Select **Commands**, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a description for this right, then define the right to switch to the service account. For example, if the service account is oracle:
    - Type su - oracle in the Command field.
    - Verify the Standard user path is selected.
6. Click the Restricted Shell tab, under Can be used in a restricted role, select **Specific user or uid**, then type root.

7. Click the Run As tab, deselect **Can be used by dzdo**.

   These settings specify that this right can only be used in a restricted shell environment and users can only run the commands that are explicitly allowed in the restricted role they are assigned. If this is the only right defined for a role, the only command users assigned to the role can run is su - oracle. For a role definition with this right to be effective, you would add command rights for the specific database operations users should be allowed to perform after switching to the oracle service account. For example, if the oracle service account is used to run a backup-all-dbs script, you would add a right to allow the execution of that script.

8. Click **OK** to use the default environment variable settings and command attributes.

   Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Define a PAM Access Right to Allow Logging On

The default UNIX Login role allows users to log on using a password or without a password in an unrestricted environment. If you are creating a role definition for a service account, you can use PAM access rights to control the specific PAM-enabled applications users can use to log on. To illustrate controlling how users log on, this example of a restricted role for the oracle service account only allows users to log on with ssh.

To define a PAM access right for a specific application:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new PAM right.
3. Expand **Authorization > UNIX Right Definitions**.
4. Select **PAM Access**, right-click, then click **Add PAM Access Right**.
5. Type a name and, optionally, a description of the PAM application for which you are adding an access right.

   For the Application field, type the platform-specific name for the PAM application as defined in the PAM configuration file or PAM directory. For example, type ssh or sshd. You can also use wildcards in this field to perform pattern matching for the application name.

6. Click **OK** to save the access right for this PAM-enabled application.

Create a Restricted Role Definition for the Service Account

After you have defined the rights that allow a user to log on using a PAM-enabled application and run the su - command for a service account, you can create a role definition for these rights. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

To create a restricted role definition for switching to a shared service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a name and description for the new role, then click **OK**.

   For example, type a name such as oracle_service and descriptive text such as Users with this role can start a secure shell session and switch to oracle.

   By default, this role is available at all times. You can click **Available Times** if you want to specify days of the week or select times of the day for making the role available.

6. Click the System Rights tab and select at least one option that allow users assigned to this role definition to log on, then click **OK**.

   In this example, users open a secure shell to switch to the service account so you might select **Non-password (SSO) login is allowed**.

   If a service account instead of a user account is used to log on, it might be mapped to a disabled Active Directory account. In this case, you might select the **Account disabled in AD can be used by sudo, cron etc** system right to ignore the disabled state and allow the service account to log on.

7. Select the new role definition, right-click, then click **Add Right**.
8. Select the rights you defined for running the switch user (su -) command and logging on with the PAM application ssh, then click **OK**.

Assign an Active Directory Group to the Role

As discussed in previous chapters, you should associate Server Suite role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership.

If you are using the recommended deployment structure and naming conventions, you would create the Active Directory group in the ou=Service Accounts, ou=Centrify organizational unit using the format ZoneName_Service_RoleName. For example, create an Active Directory group named sanfrancisco_service_oracle. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select **Role Assignments**, right-click, then click **Assign Role**.
5. Select the role definition you created for using secure shell and switching to the service account access, such as oracle_service, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role definition.
    - Select Group as the object to find.
    - Optionally, type all or part of the group name.
    - Click Find Now.

      Select the group you created for the role in the results, then click **OK**.
7. Click **OK** to complete the assignment.

Working in a Restricted Shell Environment

When users who are assigned to this role want to open a secure shell session and switch to the oracle service account, they will be placed in a restricted shell environment. Within the restricted shell, they can only execute the commands you have added to the role definition until they exit the restricted shell session. In this example, the role definition only allows users to log on using ssh and execute one command, su - oracle. If those users are also assigned the UNIX Login role, they will have access to an unrestricted shell when they close the restricted shell session.

If you want users who access a shared service account to work exclusively within the restricted shell environment, you must remove the UNIX Login role assignment in the zone or on the computer where they should only have restricted shell access. Before removing the UNIX Login role assignment, however, you should consider the trade-off between improved operational security and audit compliance and reduced operational access. Depending on the rights you add to a role that runs in a restricted shell environment, the restricted shell can dramatically limit what users can do.

Testing Access in a Restricted Shell

If you create a role definition for a shared service account that runs in a restricted shell environment, you should test it before migrating any users to it. You can use the dzinfo command with the --test option from a UNIX command prompt. For example, type dzinfo, the user name to test, the --test option, then the full path to the command to test:

dzinfo raejames --test "/usr/bin/su - oracle

You can also run the dzinfo command with the --roles option to see information about the rights defined for the current user or a specified user. For example, run the following command to check the roles and rights defined for the user raejames:

dzinfo raejames --roles

For more information about using this command, see the dzinfo man page.

What Users See in a Restricted Shell Environment

For users assigned to a role that runs in a restricted shell, logging on opens a dzsh shell. Within that shell users can only execute the commands you have explicitly defined for them. In this example scenario for a shared service account, typing su - oracle is the only allowed command. If the user types any other command, the shell reports that the command is not allowed.

Creating a Role Definition for Temporary Root Access

Another common use case for role definitions occurs when you want to provide temporary access to privileged commands. For example, you might want to provide temporary rootlevel access to an application developer troubleshooting a problem on a production server or to a consultant you've hired for a specific period of time. These types of role definitions are often used as overrides on individual computers.

The steps for creating a role definition with temporary root access are similar to the steps for creating the other roles, except that you specify time constraints for the role. The time constraints might include specific hours of the day, days of the week, or a start and end time for a role assignment. The next sections summarize the steps for creating a role with temporary root-level access.

Define a Command that Allows Root Access

The steps for defining a right for switching to the root user are similar to defining the right to run commands for the root-equivalent user, but it is recommended that you create a separate right definition for this case.

To create the right to switch to the root user:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select **Commands**, right-click, then click **New Command**.
5. On the General tab, type a name, such as emergency_access, for this command right and, optionally, a description for this right, then define the right to switch to the root user:
    - Type the command for switching to the root user. For example, type su - root in the Command field.
    - Verify Standard user path is selected.
6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

    These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.
7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

    In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.

Create a Role Definition for Temporarily Running as Root

After you have defined the right to switch to the root user, you can create a role definition for that right.

To create a role definition with the right to run the emergency_access command:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a name and description for the new role.

    For example, type a name such as emergency_access and descriptive text such as Users with this role can temporarily run commands with root

privileges.

6. Click **Available Times** to specify days of the week or select times of the day for making the role definition available.

   For example, you might want to allow access only on Friday, Saturday, and Sunday and deny access the rest of the week. After you have set the days and times for the role definition to be available, click **OK**.

7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select the emergency_access command you defined for switching to the root user, then click **OK**.

   To use this role, a user must be assigned to the UNIX Login role for the zone or a role definition that has, at a minimum, the following System Rights:

   - Password login and non-password (SSO) login are allowed
   - Login with non-Restricted Shell

Assign the Role as a Computer-level Override

In most cases, a role definition of this type is assigned to a specific computer rather than applied to all computers in a zone.

To make a role assignment on an individual computer:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer for which you want to define a computer-level role assignment.
3. Expand Computers, then select the specific computer on which you want to make a role assignment.
4. Select **Role Assignments**, right-click, then click **Assign Role**.
5. Select the role definition you created for temporary root access, such as emergency_access, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory user who should have temporary root access:

   - Leave User as the object to find.
   - Optionally, type all or part of the use name.
   - Click Find Now.

     Select the user in the results, then click **OK**.

7. Deselect **Start immediately** and set a specific Start time for the role assignment.
8. Deselect **Never expire** and set a specific End time for the role assignment.
9. Click **OK**.

Verify the Role Assignment on the Computer

You can run dzinfo --roles or dzinfo username --roles to see if the emergency_access role is available based on the start time for the role definition and the local time of the Linux or UNIX computer.

At the specified start time for the role assignment on the local computer, the user you assigned to the emergency_access role can type the following command:

dzdo su - root

The user is not prompted to provide the password and becomes the root user on the local computer until the specified role assignment end time. The one caveat to be aware of is that the user would continue to have root access after the specified end time if the shell session remains open continuously. If a user is still logged on after the time period has expired, you should check whether the user still requires root-level access. If the session has remained open but the user should no longer have root access, kill the session and log the user off.

Creating a Role Definition With Specific Privileges

The previous examples of role definitions granted broad privileges. You can also use role definitions to grant or deny very specific rights. For example, you might want to deny access to a specific set of commands for a specific group of administrators who otherwise have broad access rights or to strictly limit exactly what commands users can execute. Depending on the requirements of your organization, you might configure these types of role definitions to be used in a restricted or unrestricted shell.

The steps for creating a role definition with specific privileges are similar to the steps for creating the other roles. In this example, rights are defined to prevent the execution of specific commands and combined with a right to grant access to all commands not explicitly listed.

Define Command Rights to Prevent the Use of Commands

The steps for defining rights that deny access to specific commands are similar to the steps defining other rights, but require different syntax. In this example, you create a "blacklist" of commands users cannot execute.

To create the right to switch to the root user:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select **Commands**, right-click, then click **New Command**.
5. On the General tab, type a name, such as No password resets, for this command right and, optionally, a description for this right, then define the right:
    - Type !passwd * in the Command field.
    - Verify Standard user path is selected.

      An exclamation point (!) at the start of a command disallows matching commands. Command rights that start with the exclamation point take precedence over others that don't.
6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

   These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.
7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

   In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.
8. Repeat Step 4 to Step 7to create rights for the following specific commands:

   !groupadd *
   !useradd *
   !groupdel *
   !userdel *

Create a Restricted Shell Role Definition that Uses the Command Rights

After you have defined all of the command rights that disallow specific commands, you can create one or more role definitions to use those rights. For example, you might create one role definition to run in an unrestricted shell that requires users to invoke dzdo to execute privileged commands and another role definition that runs in a restricted shell but does not require users to execute privileged commands using dzdo. The second role might be useful if you have existing scripts that would have to be modified if invoking dzdo is required.

To create a role definition for specific command rights:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.

3. Expand Authorization.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a name and description for the new role.

   For example, type a name such as operators and descriptive text such as Users with this role can run privileged commands but not reset passwords, add or delete users and groups.
6. Click **System Rights** if you want this role definition to be used in a restricted shell environment as a replacement for the predefined UNIX Login role.

   To use this role, a user must be assigned to a role definition that has at least one UNIX system right, such as Password login and nonpassword (SSO) login are allowed or Nonpassword (SSO) login is allowed.
7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select all of the command right that disallow specific operations, the command right that grants access to all remaining commands, and a PAM access right, then click **OK**.

   For example, you might add the following previously-defined command rights to this role definition:

   No password resets
   No user adds
   No group adds
   No user deletes
   No group deletes
   Root like access (* for all commands not explicitly disallowed)
   PAM ssh/login allowed

   This role definition allows members of the operators role to execute any command within a restricted shell environment except those explicitly disallowed, including privileged commands, without invoking dzdo first. You can assign the role definition to the appropriate Active Directory users or groups like the previous role definitions.

Create an Unrestricted Shell Role Definition that Uses the Command Rights

The command rights were configured to allow execution in either a restricted shell environment or an unrestricted shell environment. In an unrestricted shell environment—for example, the default shell environment when users are assigned the UNIX Login role—commands that require administrative privileges must be executed by first invoking the dzdo command, which is similar to invoking commands with sudo.

You can control whether users are required to enter a password when they execute privileged commands using dzdo by setting the **Authentication required** on the Attributes tab when you create a command right. By default, no password is required. If you were adding a new command right that requires authentication, you would click the Attributes tab, select **Authentication required** then select one of these options:

- **User's password** if users are required to enter their own password before executing the command.
- **Run as target's password** if users are required to enter the password for the target account that is executing the command.

In most cases, the default of no password is appropriate because the user has been previous authenticated before invoking dzdo to execute a privileged command and the **Run as target's password** option requires the user to know the privileged account password. For example, if the run-as user is root, the **Run as target's password** authentication option requires the user to know the password for the root account.

The steps for creating the role definition that includes the previously-defined command right are the same for the unrestricted shell as for the restricted shell except that at Step 6 of Create a restricted role definition for the service account in the System Rights tab, you would also select the **Login with non-Restricted Shell** option if you are not using the UNIX Login role. You could add all of the same command rights to the role definition and grant the same privileges and exceptions.

The primary difference between the two role definitions would be how users execute their privileged commands.

In the restricted shell environment, users running the adflush command requiring administrative privileges:

dzsh $ adflush

In the unrestricted shell environment, users running the adflush command requiring administrative privileges:

[tulo@ajax]$ dzdo adflush

Creating a Role Definition with Rescue Rights

The Rescue rights option allows you to control which users should be able to log on if problems with the authorization cache or auditing service are preventing all other users from logging on. For example, if you have a computer with sensitive information, such as credit card numbers or intellectual property, you might require auditing for all users in the role with access that computer. If the auditing service is stopped or removed on that computer, no one would be able to log on and use the computer until auditing is restored. If you create a role with the Rescue rights option selected, only the users assigned to that role are able to log on and continue working until the problem that caused the lockout is found and fixed.

Users who are in a role granted access because they have rescue rights can still be audited through the system logging facility. However, their activity is not recorded in the audit store database if the auditing service is not available.

Creating Additional Custom Roles and Role Assignments

The previous sections described common roles that organizations implement to begin the process of migrating and removing locally defined privileged accounts. For most organizations, locallydefined accounts with privileged access present a security risk and are often identified as a compliance issue by auditors.

By creating role definitions similar to those described in this chapter, you can eliminate the need to share root and service account passwords while still providing privileged access to computers where it's needed. These additional roles are not required, however. You can choose to create them or create a completely different set of role definitions to suit your organization. For example, you might decide to create custom roles specifically tailored to the needs of database administrators, backup operators, and web application developers. Similarly, you might decide to create separate role definitions that are customized with AIX command rights for AIX administrators that are different from the command rights defined for Solaris administrators.

As with the common role definitions, additional custom role definitions can be created in the top-level parent zone and available throughout the zone hierarchy or in any child zone. They can also span all the computers in a zone or be assigned specifically to individual computers.

If you plan to create your own custom role definitions and role assignments, keep the following key points in mind:

- Rights associated with roles are cumulative. Users receive all of the rights in all of the roles they are assigned.
- Users must be assigned at least one role that allows an interactive login or Kerberos authentication to have any access to any computers. For existing users, this is accomplished by assigning the default UNIX Login role during the migration to Active Directory.
- Users must be given the Login with non-Restricted Shell system right to have access to a full shell. If they assigned in a role without this right, they can only execute the commands explicitly defined for their role.

For users who have previously had full shell access, this limitation can be frustrating, unexpected, and unworkable. Before placing or moving users into a restricted role, be sure those users and managers throughout the organization are well-informed and wellprepared for the change and understand the business reasons for the change.

Working with Computer Roles

In addition to the role definitions that confer specific rights when assigned to users and groups, Server Suite provides a mechanism for linking a specific group of computers to a group of users with a specific role assignment. These computer-based access rules, called **computer roles**, identify computers that share a specific attribute that you define and a set of users with common access rights.

For example, you can define a computer role that identifies a set of computers as Oracle database servers linked to a set of users who have been assigned the oracle_dba role. You can then add and remove users from the Active Directory role group linked to the oracle_dba role to grant or remove the rights associated with the oracle_dba role. In this example, the computer role identifies computers that host Oracle databases and the set of users assigned the database administrator role.

The same set of computers might include computers with AIX and Solaris operating systems. You could then create separate computer roles that link the AIX computers to a group of AIX administrators and the Solaris computers to a group of Solaris administrators.

Planning to Use Computer Roles

Because computer roles provide you with a great deal of flexibility for defining access rights, you might want to do some planning before you create new computer roles. For example, before you create a computer role you must know the criteria you want to use to group computers into one or more Active Directory security groups. You must also identify the users who will have a common set of access rights based on the computer grouping.

At a high-level, defining a computer role requires the following:

- Identify computer roles you want to define.

  Decide on the attribute the computers in a particular group share. For example, you can use a computer role to identify computers in the web farm, that host specific applications, or serve a specific department.

- Identify the users for the computer role and create Active Directory groups for them.

  You might need multiple groups because different sets of users have different access requirements. For example, if you are creating a computer role for a set of Oracle servers, you might need separate Active Directory groups for database users, database administrators, and backup operators.

- Identify the role definitions each set of users should be assigned.

  You might need to create specific access rights and role definitions for different sets of users. For example, if you are creating access rights for database users, database administrators, and backup operators, the database users may be able to use the predefined UNIX Login role, while administrators need permission to run privileged commands, and backup operators might be assigned a limited set of commands in a restricted shell.

How Computer Roles Simplify the Management of Access Rights

Deciding how best to use computer roles requires some upfront planning and configuration that might not be part of your initial deployment plan. To make effective use of computer roles, you also need to plan for and prepare appropriate role definitions for different sets of users. However, computer roles provide a powerful and flexible option for managing access to Server Suite-managed computers using your existing processes and procedures for managing Active Directory group membership.

For example, if you create a computer role group for Oracle servers and you deploy a new Oracle server, you simply add the computer account for the new server to the computer role group in Active Directory. If new database administrators join your organization, you simply add them to the Active Directory security group for Oracle database administrators. The computer role links the computer role group to the user role assignment and no additional updates are needed to accommodate organizational changes. If you need to modify the access rights, you can change the role definition and have the changes apply to all members of the group.

Because creating and managing computer roles is typically an ongoing administrative task after initial deployment, it is covered in the *Administrator's Guide for Linux and UNIX*. <!--TODO: xref -->

Migrating And Managing Service Accounts

After you have migrated accounts for the users who log on to the UNIX computers in your organization, the next step in the deployment is to decide how you want to manage the service accounts for applications in your environment. This section describes the options available and why you should consider migrating local service accounts to Active Directory.

Why Migrate Service Accounts?

A service account is typically a local user and group account dedicated to a specific application or to performing specific operations. In many cases, the service account has escalated permissions that allow it to run privileged operations on behalf of the application it supports. In addition, service accounts often have no password or a password that is wellknown to multiple users. Service accounts without a password typically require a local sudoers policy to control access. Service accounts with a shared passwords present a security risk because users can avoid an audit trail and, if passwords are managed locally, they may not conform to the password policies that are enforced for normal user accounts.

Therefore, the primary reason for migrating service accounts to Active Directory is to provide better security for accounts that can execute privileged commands, start and stop processes, or run specialized jobs on computers in your network.

Note that not all organizations choose to migrate and manage service accounts in Active Directory. There is no technical requirement that you do so. However, Server Suite provides you with several options for improving the security for service accounts. You should consider the options available, then decide which, if any, are most applicable for your environment.

Identifying Service Accounts to Migrate Tto Active Directory

Every UNIX platform has its own set of standard service accounts that are installed by default. For example, most UNIX platforms include services accounts for common applications, such as gopher, mail, ftp, and uucp. For most of these standard service accounts, there's no business reason to map them to accounts in Active Directory, unless you are trying to eliminate all local accounts on your UNIX computers.

In most cases, you can skip migration for the standard service accounts included by default when you installed the operating system as described in Eliminate default system accounts.

However, service accounts that run or manage applications or own an application's files are typically good candidates for mapping to Active Directory users. For example, an Oracle database instance has an oracle service account that owns the database server and the related processes that run in the background. Although usually linked to an application, a service account can also be account created to run scheduled jobs and own the files related to those jobs.

Service accounts that are good candidates for mapping to Active Directory users are ones that perform business operations without a password, rely on a shared password known to multiple users, or use shared SSH keys.

Service Accounts Without a Password

Most UNIX service accounts do not use passwords because UNIX services don't require an interactive log on to own files or run jobs. The most common way for users to access the service account is through the configuration of the sudoers file. The sudoers file provides rules that allow a subset of users to run the su command and change to the service account user. Mapping this type of service account to an Active Directory user eliminates the need for managing access through local sudoers policies and enables you to enforce the same password complexity rules for service accounts as normal user accounts.

Service Accounts with a Shared Password

The second most common way for users to access service accounts is with a shared account password. In this scenario, multiple users know the password for the service account and may be able to log on directly as that account. With shared accounts, there is no authoritative way to identify who is logging in to use the account. If you have any service accounts that rely on a shared password, you should consider migrating those accounts to Active Directory to eliminate the shared password.

Service Accounts that Use SSH Keys

Another common attribute of service accounts is that they often have a set of SSH keys that are available on multiple computers. The SSH keys allow the service account to transfer information from one UNIX computer to another without a password. In this scenario, a specific or the default SSH key for the service account exists in the authorized keys file on each of the computers to which the service account must connect.

Mapping a Service Account to an Active Directory User

After you identify one or more service accounts as candidates for mapping to an Active Directory user principal, you should identify an appropriate Active

Directory user principal for the service account to map to. In most cases, there won't be an appropriate user already defined in Active Directory, so you will need to create one or more new users.

Create a New Active Directory User Account

You can use Active Directory Users and Computers or another tool to create a new user principal for each service account you are migrating to Active Directory.

> **Note**: In most cases, you submit a request for a new account to be created using the procedure defined for your organization. For example, you might submit a request by filling out a service desk ticket and have the request serviced by a member of the account fulfillment team. The steps in this section only apply if you have permission to create new Active Directory user accounts. If you are not responsible for creating new Active Directory user principals, you can skip the following procedure.

To create a new Active Directory user for a service account:

1. Start Active Directory Users and Computers.
2. Expand the forest domain and the top-level UNIX organizational unit you created in Selecting a location for the top-level OU.
3. Select Service Accounts, right-click, then select **New > User**.
4. Type a name and account login information for the service account, then click **Next**.
5. Type and confirm the password to use for the service account in Active Directory, select the **User cannot change password** and **Password never expires** options, then click **Next**.

   The password must conform to your existing password policies for Active Directory users.

   If you are creating a new user to replace a shared account, type the password currently in use if it is acceptable within your site's Active Directory rules for password complexity. If you use the shared password, you should change the password after migration. If the current password is not complex enough, you should type a new password that complies or contact the Active Directory Enterprise Administrator for alternatives.
6. Click **Finish** to complete the creation of the new user principal.

Map the Unix Service Account to the Active Directory User

After you create a new Active Directory user principal for the service account, the next step is to map the UNIX account to the Active Directory user.

In preparation for this step, you should notify the user community that the service account will be unavailable for a brief period of time, so that you can make the change and verify that everything works as expected. You should then stop the service and any jobs associated with the service account.

By notifying users and making the service account unavailable for a period of time, you can prevent the change from affecting people who depend on the service to do their jobs. You can then use Access Manager to select the service account and map it to an Active Directory user.

To map the service account to an Active Directory user with Access Manager:

1. Start Access Manager.
2. Navigate to the UNIX user account
3. Navigate to the service account under a specific computer's Users node or under the Local Account Users node.
4. Select the service account, right-click, then click **Map to AD User**.
5. Type all or part of the Active Directory user name, click **Find Now**, then select the account in the results and click **OK**.

   Clicking OK updates the configuration on the remote host. You could accomplish the same thing by manually editing the configuration file (centrifydc.conf) or with a group policy.
6. Verify the service starts and executes operations as expected by switching to the root user or the service account and attempting to start the service.
7. Check for messages in the log files that the service account writes to. The entries should be regular service startup messages. You should verify that there are no errors or authentication failure messages.

After you verify that the service starts as expected and that any jobs it owns start successfully, you can notify users that the service is available or do additional testing. Depending on your organization and the service account you have mapped to Active Directory, developers, database owners, application owners, and others may want to do full regression testing or execute specific test cases.

How the Mapped User Changes Your Environment

The Active Directory user you create for a service account must be enabled for authentication to work. However, enabling this new user account does present some potential risks to your environment. For example, on UNIX computers, creating the new user may have added a password for a service account that did not previously have one. If the new password is known to more than one person, the account may be considered a shared account and result in an audit finding.

Also, because the new account is a valid Active Directory user principal, anyone with the password can potentially log on to any Windows computer in the forest. By giving the service account a valid password and enabling the account, you have granted access to the Windows network for an account that previously had no access to Windows computers.

If you disable the account, you prevent that account from accessing all Windows and UNIX computers, running jobs, or executing required tasks. If you leave the account enabled and the password is compromised, both Windows and UNIX computers are vulnerable to attack. Even if the password is not compromised, failed password attempts could trigger an account lockout policy, rendering the service unusable.

Mapping service accounts to Active Directory users is a simple technique for managing access and password complexity for service accounts. If you have strong passwords and carefully control access to the account and its password, you can mitigate the risks. This strategy is also best suited to service accounts that already use a password. However, if granting the service account access to the Windows network presents too great of a risk, you should consider alternatives.

Creating a Service Account Role in a Zone

As discussed in How the mapped user changes your environment, mapping a service account to a Windows user makes the account vulnerable to attack. If the attack results in a guessed password, the attacker would be able to log on as the service account, and, potentially, impersonate the service account on multiple computers on the network using SSH keys. Because the mapped user is also a valid Windows account, a successful dictionary attack might also grant access to Windows computers on the network. If the attack did not result in a guessed password, the failed password attempts could lock out the service account, making it unusable.

For service accounts that do not have a password, this vulnerability to a password-guessing attack would be a new security risk that did not previously exist. Therefore, simple account mapping is typically not the best solution for service accounts that are secured using sudoers policies or SSH keys instead of an account password.

If simple account mapping is not the appropriate solution for the service accounts in your organization, you may want to consider creating one or more service account roles. Roles enable you to securely manage the privileges of UNIX service accounts through Active Directory.

Create an Active Directory User Account for the Service

Because roles are tightly integrated with Active Directory user and group definitions, linking a service account to a role requires that you have an Active Directory user object to work with. In most cases, you should use your existing procedures to request a new user account. If you are responsible for creating new Active Directory user principals, see Create a new Active Directory user account.

Define a New Role with System Rights

It is recommended that you create the role definition for a service account in the appropriate child zone. If you want to make the role definition available in all child zones, however, you can create it in the parent zone. The specific selections you make for the role depend on the requirements of the service account for which you are creating the role definition. The steps described here provide general guidelines. Other settings may apply for the role definition in your organization.

To create a new role for a service account:

1. Start Access Manager.
2. In the console tree, expand **Zones** and the top-level parent zone.
3. Select the specific zone for which you want to define a role, and expand **Authorization**.
4. Select **Role Definitions**, right-click, then click **Add Role**.

5. On the General tab, type a name and description for the new role, then click **OK**.
6. Click the **System Rights** tab and select the following options that allow the service account to access UNIX computers using SSH keys or Kerberos, then click **OK**:
    - Non-password (SSO) login is allowed
    - Account disabled in AD can be used
    - Login with non-restricted shell

    In most cases, you should select the Login with non-restricted shell option. This option enables the service account to execute all of its commands in a standard shell. To have the service account run in a restricted shell, you must be able to identify and define rights for all of the commands that the service runs. The service account must also be able to execute all of its commands within the restricted shell (dzsh) environment. For most organizations, this additional security requires significant research and testing before it can be implemented. However, forcing a service account to run in a restricted shell reduces the likelihood that a compromised service account could be used to attack computers on the network.
7. Select the new role, right-click, then click **Add Right**.
8. Select the login-all right for the zone, then click **OK**.

    This predefined right grants access rights for all PAM applications. If you determine that a specific service account should only use a specific PAM application, such as SSH or FTP, you can define a right that only allows that application to be used, then select that right in the role definition to specify that the service account must use the selected PAM application for access.

Create a Unix Profile for the Service Account and Assign the Role

After you define the role for the service account, you can create a UNIX profile for the service account. In most cases, you should define the UNIX profile for service accounts using machine-level overrides, rather than defining them for zones. Defining profiles for service accounts using machine-level overrides has the following advantages:

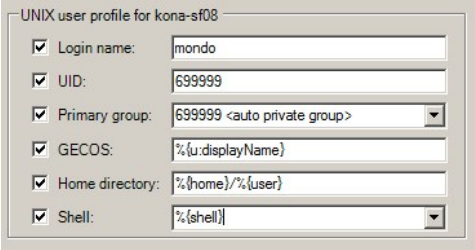
- Profile attributes are not affected the Zone Provisioning Agent. Most service accounts require specific UID and GID values. By specifying these values using a machine-level override, you don't have to worry about them changing when the Zone Provisioning Agent runs.
- You can explicitly identify which computers the service account can run on. If you define the UNIX profile for a service account in a zone, all of the computers in the zone are available to the same service account. If you define the UNIX profile using machinelevel overrides, the service account only runs on computers where it has a profile and the profile attributes for the service account can be different on different computers in the same zone.
- You can restrict the scope of the role assignments on a computer-by-computer basis. By defining the UNIX profile using machine-level overrides, you can configure different service account owners for development, testing, and production computers in the same zone.

If the profile attributes are consistent across most of the computers in a zone and the service account should run able to run on all of those computers, you can define all or part of the UNIX profile for the parent or child zone to reduce the management of profile attributes on individual computers. However, if the legacy accounts had different attributes on different computers, it is typically best to use machine-level overrides.

To create the UNIX profile and assign the service account role using machine-level overrides:

1. Start Access Manager.
2. In the console tree, expand **Zones** and the top-level parent zone.
3. Expand the Child Zones node, select a specific child zone and expand it to display the Computers node.
4. Select computer for which you want to define machine-level overrides, right-click, then click **Add User**.
5. Click **Browse** to search for and select the Active Directory user account for the service, then click **Next**.
6. Select **Define user UNIX profile** and **Assign roles**, then click **Next**.
7. Select and define the attributes in the UNIX profile for the service account, then click **Next**.

    You can use inherited default values for any of the attributes from the default values specified for the zone or selectively override the default values for any of the attributes. For example if you define user defaults using runtime variables in the zone, you can use the inherited values for the Login name, GECOS field, Home directory, and Shell and explicitly define the UID and primary GID for the service account profile.



8. Select the default UNIX Login role, click **Remove**, then click **Add**.
9. Click **Browse**, select the role you created for the service account, then click **OK**.
10. Click **OK** to accept the default start and end times for the role assignment, then click **Next**.
11. Review your selections, click **Next**, then click **Finish**.

Secure the Active Directory User Account

At this point, you have an enabled Active Directory user mapped to a UNIX profile for the service account. Having an enabled Active Directory user mapped to a service account, however, still presents a potential security risk as discussed in How the mapped user changes your environment. The next step is to decide how to secure the account to reduce the risk that it will be compromised and allow an attacker to gain access to the computers on your network. Depending on your organization's infrastructure and requirements, there are essentially two options available:

- Use SSH public key authentication
- Use Kerberos authentication

Each of these options has advantages and disadvantages and require different steps to configure.

Using Ssh Keys for Authentication

If you already use distributed SSH keys on hosts that run services, you can take advantage of that infrastructure and secure the service account by disabling the Active Directory user object in Active Directory. This is a common configuration that enables computertocomputer communication without a pass-phrase.

To use SSH public key authentication:

- Define the role for the service account with the **Account disabled in AD can be used by sudo, cron, etc** system right enabled. The role should not allow an interactive login.
- Ensure that the computers that communicate with each other have the SSH public key in the authorized_keys file.
- Select the Active Directory user principal you created for the service account and set the **Disable Account** option.

After you disable the account, it cannot be used for authentication on Windows or UNIX computers and it is not susceptible to a password-guessing attack. The account can continue to run UNIX services and use PAM-aware applications to communicate to other computers on the network using the SSH public key.

The primary advantage of using SSH authentication is that if you already have SSH public keys distributed to allow computer-to-computer communications, services should continue to work after the Active Directory user principal is disabled. There is very little configuration required to implement this solution.

There are, however, disadvantages to using SSH public keys. For example, you must manage key distribution. To allow a UNIX service account to communicate with other UNIX computers, you must generate the SSH key, and distribute that key to all of the hosts that the service account needs to communicate with. In addition, the most common configuration of SSH authentication allows keys to be used without a pass phrase. If the private key is compromised, an attacker could effectively impersonate the service account across multiple computers on the network and reacting to a compromised key can be timeconsuming because it requires you to remove the public key from every $HOME/.ssh/authorized_keys file distributed across the enterprise.

Using Kerberos Tickets for Authentication

If you are not already using distributed public-private SSH keys for authentication, you may want to consider using Kerberos authentication. Kerberos authentication is more secure than SSH keys and using Kerberos enables you to centrally manage access for service accounts, but it requires additional configuration.

To use Kerberos to secure UNIX service accounts:

- Install Kerberos-enabled software. You can use the Server Suite-provided version of OpenSSH or OpenSSH, version 3.9 or later, if it has been compiled with Kerberos support.
- Ensure the Active Directory user principal for the service account is enabled. Unlike SSH authentication, Kerberos requires the user account to be enabled to request ticket granting tickets or service tickets for other computers.
- Use the setspn.exe program to create at least one new Service Principal Name (SPN) for the UNIX service account.
- Run the adkeytab command on every UNIX computer where you want to re-use the Active Directory user principal that you created the new SPN for. This command creates a Kerberos keytab file that is only readable to the service account user. The keytab file allows the service account to request a ticket granting ticket so that it can communicate with other UNIX computers.
- Use the kinit command to request a ticket granting ticket from Active Directory for the UNIX service account. The ticket granting ticket (TGT) allows the service account to request additional host tickets for SSH communications to other UNIX computers.
- Use the klist command to list the tickets for the UNIX service account.

Testing And Migration

If you decide to use Kerberos for service accounts, you should test the computertocomputer communication. If you are migrating from authentication using SSH public-private keys, you can move ore rename the authorized_keys file for the service account on remote hosts to test authentication. After you test the Kerberos authentication of SSH communications and are certain that it works, you can delete the id_rsa, id_rsa.pub, and authorized_keys files for the service account that you have migrated.

Renewing Ticket Granting Tickets

Using Kerberos gives you consolidated control over UNIX service accounts. If an account is disabled in Active Directory, it cannot be used after any existing tickets expire.

If the service account runs scheduled jobs, you may want to create a crontab entry for the UNIX service account to run the kinit command at a regular interval so that the TGT doesn't expire. If the ticket expires and the kinit command isn't embedded in the job the service run, computer-to-computer communication will fail until the next time kinit is executed. For example, you may want to add logic to run klist to check whether there is a valid TGT, then run kinit is no valid TGT is found.

More Information

For information about using the setspn.exe program, see the Microsoft documentation for that program. For information about using Server Suite command line programs, see the corresponding man page.

Remove Local Service Accounts from Remote Computers

After you have migrated service accounts to roles in Active Directory, tested operations, and verified that commands and jobs run as expected, you can remove the local service accounts from remote computers to prevent them from being used.

For most organizations, removing local service accounts is a recommended security practice. However, you should leave the default operating system accounts as local accounts. In most cases, those accounts are not migrated to Active Directory. The default accounts for each platform are listed in the user.ignore file that is installed with the platform-specific Server Suite Agent.

Planning to Deploy in a Demilitarized Zone (DMZ)

Many organizations require both an internal network for corporate assets and a physical or logical subnet that exposes resources or services to a larger, external network, such as the Internet. For security, computers and resources in the external-facing perimeter network or demilitarized zone (DMZ) have limited access to the computers in the internal network. Because communication between the computers in the DMZ and the corporate network is restricted and protected through the use of a firewall, there are specific constraints on configuring authentication and authorization services.

This section describes how to deploy Server Suite components in specific DMZ scenarios.

Identifying the Computers to Protect

The computers that are most vulnerable to attack are computers that provide services such as e-mail, host external-facing web applications, and manage network routing through Domain Name Servers (DNS). Computers that provide these services are typically isolated from the internal network on their own subnet and allowed to communicate with the internal network through specifically designated channels. This configuration allows computers in the DMZ to provide services to both the internal and external network, but controls the traffic allowed to be routed between the computers in the DMZ and the internal network clients.

Creating a Forest and Trusts for a DMZ

It is recommended that you create a separate Active Directory forest for the computers to be placed in the network segment you are going to use as the demilitarized zone. You should then establish a **one-way outgoing trust** from the internal forest to the DMZ forest.

Defining a one-way trust allows existing internal forest users to access resources in the DMZ without separate credentials or being prompted for authentication. The one-way trust also prevents any accounts defined in the DMZ forest from having access to the internal network. Accounts defined in the DMZ forest can only access computers inside the DMZ domain. If a privileged account in the DMZ forest is compromised, that compromise is limited to the scope of the DMZ forest.

For Server Suite, the one-way trust enables you to:

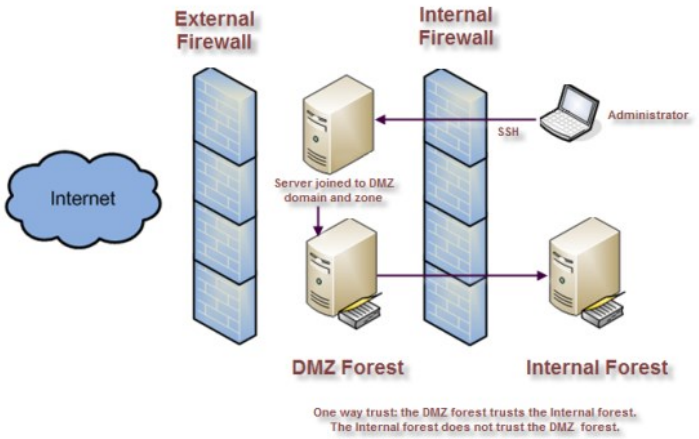
- Use the internal forest for authentication and authorization services for user accounts.
- Define computer accounts in the DMZ domain without permission to read data from the trusted domain of the internal forest.

In most cases, you should not use an existing Active Directory domain when deploying Server Suite Agents in a DMZ. Using an existing domain requires opening additional ports through the internal firewall to allow computers to connect directly to the domain controllers in the internal forest. Allowing computers in the DMZ to connect directly to the internal forest implicitly grants access to resources behind the internal firewall.

Defining Zones for Computers in the DMZ

You should create one or more zones in the DMZ forest and specify those DMZ zones when computers join the DMZ domain. You should also create and manage UNIX group profiles in the DMZ domain. However, you should define user accounts and UNIX profiles for in the internal forest and not the DMZ forest. Defining user accounts in the internal forest ensures that users can use a single password to authenticate across all network resources, including those in the DMZ.

The following figure provides an overview of the basic architecture for deploying Server Suite Agents when you have computers in a DMZ.



To enable authentication for resources in the DMZ, users must have:

- A valid Active Directory user principal.
- A complete UNIX profile in one or more zones in the DMZ.
- A UNIX Login, listed, or custom role assignment that allows access computers in the DMZ.

Creating a Firewall and Securing the Network

You should establish firewall rules that allow communication from the DMZ domain controllers to the internal domain controllers. Other computers in the DMZ, for example, the UNIX servers you want to isolate from the internal network, should have limited access to the corporate network. The most common exception is to allow communication from the corporate network to the DMZ network using port 22.

The client and server port requirements to enable communication through the firewall depend on the Windows operating system you have installed on the domain controllers and the functional level of the forest. For information about the specific ports to open and the services that use the ports, see Microsoft Active Directory documentation, such as *How to configure a firewall for domains and trusts*.

In addition to configuring a firewall that minimizes exposure to the corporate network, you should remove insecure network protocols, if possible, or replacing insecure authentication programs, such as POP3 and FTP, with Kerberos-enabled programs. These security steps reduce the potential for password exposure and the risk of an account in the corporate domain being compromised.

How to Join a Domain with a Read-Only Domain Controller (RODC)

With Windows Server 2008, you have the option of installing read-only domain controllers (RODC). Read-only domain controllers enable you to deploy a domain controller that hosts read-only partitions of the Active Directory database. Deploying a read-only domain controller enables you to make Active Directory data and reliable authentication services available in locations that cannot ensure physical security required for a writable domain controller.

You can also deploy read-only domain controllers to handle special administrative or application management requirements. For example, you may have line-of-business applications that are required to run on a domain controller, or application owners who must have access to the domain controller to configure and manage operations but not allowed to modify Active Directory objects as they could with a writable domain controller. You can grant a non-administrative domain user the right to log on to the read-only domain controller while minimizing the security risk to the Active Directory forest.

For more information about read-only domain controllers, see the *Read-Only Domain Controller (RODC) Planning and Deployment Guide*.

To join a domain that has a read-only domain controller:

1. Create a computer account for the computer in the DMZ that will connect to the readonly domain controller using a writable domain controller as described in Creating computer objects for the target set of computers.

   Note: You can create the computer account using the Access Manager console, an adedit script, or using the adjoin command with the --precreate command line option. However, be sure to create the computer account in a DMZ zone.

2. Use Active Directory Sites and Services or the repadmin program to replicate the computer account in the read-only domain controller. For example,
   - In the console tree, expand Sites, and then expand the site of the domain controller that you want to receive configuration updates.

- Expand the Servers container to display the list of servers that are currently configured for that site.
- Double-click the server object that requires the configuration updates that you want to replicate.
- Right-click **NTDS Settings** below the server object, and then click **Replicate configuration to the selected DC**.
- In the Replicate Now message box, click **OK**.

3. (Optional) Open a Command Prompt and use the repadmin /showrepl command to verify successful replication on the read-only domain controller.
4. Block the route from the UNIX computer to the writable domain controller, if necessary.
5. Run the adjoin command with the self-service option. For example:

   adjoin mydomain.local --password c%ntrify --name quad90 --selfserve

   Because you have already created the computer account in Active Directory, you don't need to specify the zone to join the domain.

Enabling NTLM Authentication through a Firewall

Having a domain controller in the perimeter forest trust the internal domain requires you to open up ports through the firewall. The specific port requirements depend on the Windows operating system version and functional level of the forest. As an alternative, you can use NT LAN Manager (NTLM) authentication to allow Active Directory users in the internal forest to log on to computers in the perimeter forest.

Using NT LAN Manager (NTLM) authentication enables you to have a more restrictive firewall with a one-way forest trust between the perimeter forest and the internal forest. For example, if the firewall prevents you from using the ports required for Kerberos authentication or if you have limited communications between the forests to a specific port, you can use NTLM authentication to pass authentication requests from the domain controllers in the perimeter domain to the internal domain controllers through the transitive trust chain.

> **Note**: This configuration still requires a one-way trust relationship between the internal forest and domain controllers outside of the firewall.

Configuring the Domain Controllers that Allow NTLM Authentication

You can use the pam.ntlm.auth.domains configuration parameter to specify the domain controllers in the DMZ forest that should use NTLM authentication. This parameter requires that you specify the domain controllers using their Active Directory domain names. In addition to setting this parameter, you must be able to map NTLM domain names to their corresponding Active Directory domain names to support looking up user and group information in the cache.

Configuring a Map that Converts NTLM Domains to Active Directory

For Server Suite to automatically construct this map, it must be a able to send LDAP search requests to the domain controllers in the corporate forest. If the firewall restrictions will block these search requests, you must manually define a topology map that converts NTLM domain names into Active Directory domain names. To manually configure how Active Directory domain names map to NTLM domain names, define entries in the /etc/centrifydc/domains.conf file using the following format:

ActiveDirectory_Domain_Name: NTLM_Domain_Name

For example:

arcade.com: ARCADE
ajax.org: AJAX

You can refresh the list of domain controllers in DMZ forest at any time by modifying the configuration parameters, then running the adreload command.

Managing and Evolving Operations After Deployment

In previous sections, you prepared for deployment, migrated existing users, configured automated provisioning for new users and groups, and added one or more custom roles for privilege management. Most of these activities are related to the initial deployment and extending deployment to additional sets of target computers and interactive users.

This section discusses management activity, evolving operational security, and adding services to extend authentication and authorization after deployment. Often, at this stage, the deployment project team begins to transition activity to an operations team or support staff.

Understanding How Server Suite Software Affects Operations

Through Active Directory, Server Suite software provides a consolidated solution to authentication, authorization, and policy management for Linux, UNIX, and Mac OS X computers. Because of this consolidation, however, you may need to make changes or additions to the IT tasks or operational procedures you currently have in place. Therefore, when you deploy Server Suite software in a production environment, you should consider how it impacts the management tasks typically performed by operations staff members.

The routine tasks that may be affected by adding Server Suite software to the environment fall into the following categories:

- Change management
- System administration
- Security administration
- Service desk operations
- Capacity management

Understanding Change Management Activities

Change management typically involves testing and installing updates to the operating system or installed applications. For example, most organizations follow a controlled process for reviewing and implementing changes to the operating system because of patches or new releases.

If you are preparing to update the operating system, the support staff should also plan to test that user log-ons and role assignments continue to function correctly after update. If a system patch or update affects the operation of Server Suite software, you should contact Customer Support to determine whether the patch is supported.

Staff members should also periodically review new and maintenance releases of Server Suite software to get the latest features, fixes to known issues, and enhancements requested by Server Suite customers. After downloading the software, you can review the release notes included in each package to determine what's changed and the suitability of the update for your organization.

Understanding System Administration Activities

Most system administration tasks involve managing users and groups. After deploying Server Suite software, this information is centralized in Active Directory for both Windows and non-Windows computers. Therefore, any administrative action for a user account affects that user on both Windows and UNIX computers. For example, if you disable a user account in Active Directory, the user will be unable to log on to any Windows or UNIXbased computer in the forest.

Typically, there is a period of time where staff members must use one set of steps for provisioning users and groups on the computers not yet joined to the domain and another set of steps for provisioning users and groups on computers that have successfully joined the domain. After you complete the migration to Active Directory, you can leverage the processes and tools you use for provisioning Windows users and groups for ongoing administration of UNIX users. For example, you can use Active Directory Users and Computers, in-house custom scripts, Access Manager, ADEdit, or another management tool to perform administrative tasks.

After deployment, you should prepare any site-specific or platform-specific instructions the operations staff should follow if you are making changes to the processes or tools they are familiar with.

Understanding Security Administration Activities

Security administration involves ensuring that operators are granted the appropriate rights for administering the computers and attributes that are required as part of their job but are prevented from accessing or changing computer settings outside their areas of responsibility.

Delegated Administration

Server Suite enables administrators to explicitly delegate management tasks to the appropriate users and groups on a zone-by-zone basis.

Password Policy Enforcement

Server Suite enables you to use existing Active Directory password policy rules, such as the minimum password length, complexity requirements, expiration, and allowed number of logon failures to allow before locking an account for both Windows and UNIX users.

Privileged Account Management

Server Suite enables you define rights, roles, and role assignments to control what users can do and who can execute privileged commands. You can also map privileged local accounts to Active Directory accounts to ensure better password security for those accounts. For example, the local root user account has full access to all data and can manipulate all settings on a UNIX computer. Mapping the local root user to an Active Directory user account enforces the Active Directory password policies on the account and makes it more difficult for an unauthorized user to obtain escalated privileges on the UNIX computer.

Understanding Service Desk Operations

Active Directory and Server Suite simplify help desk and service desk operations by:

- Enabling centralized administration for tasks such as adding new users or granting access to new computers.
- Consolidating user passwords and reducing the need for password resets.
- Simplifying troubleshooting for log on failures for UNIX users.

You should provide help desk staff with troubleshooting instructions to help them diagnose and resolve failed authentication or authorization tickets.

Understanding Capacity Management Activities

During the deployment of Server Suite Agents, you should monitor and analyze network traffic and domain controller replication to determine how well your environment handles the extra load of UNIX users and computers in Active Directory. In general, Server Suite software is configured to use minimal system resources and network bandwidth. In practice, however, you should monitor and evaluate the volume of traffic to determine its impact on performance across the network and the performance experienced by users logging on to UNIX workstations and servers.

If the network traffic or resource usage exceeds your expectations, you may want to modify the default configuration to better suit your network topology. For example, Server Suite provides numerous group policies and configuration parameters that you can modify to optimize network activity or control how much data is stored locally on individual computers.

Determining Whether You Need More Resources

In most cases, deploying Server Suite software does not noticeably affect the performance of the network or domain controllers. However, if you have a widely distributed network or replication delays, you should analyze your network's capacity to handle the additional load of UNIX users and computers to determine whether you need to make changes to ensure optimal performance and availability. For example, the following factors may require you to allocate additional resources:

- If the UNIX computers are in a different physical location than the domain controllers that they access, you may want to install a domain controller on a computer that is physically closer to the UNIX computers to reduce long-distance network traffic and the chance of replication delays.
- If you need to ensure availability in the event of a network or server failure, you should ensure that you have an adequate number of domain controllers to support the UNIX computers when they need to fail-over to a backup domain controller.
- If you add a large number of UNIX users to the Active Directory domain, apply your standard method for balancing domain controllers per number of users.
- If you add a large number of UNIX computers to the Active Directory domain, apply your standard method for balancing domain controllers per numbers of computers.
- If you move a large number of UNIX users and groups from a local directory (/etc/passwd and /etc/group) to Active Directory, you may need additional network bandwidth because authentication and authorization requests are now done over the network.

For more information about modifying configuration parameters, see the *Configuration and Tuning Reference Guide*.

Understanding How Caching Facilitates Lookups

Server Suite Agents store credentials in a local cache to reduce the network traffic required to look up information in the directory. For example, if a user executes the directory listing command in a UNIX command shell (such as with the ls -l command), the command looks up and displays a listing of files along with their attributes, such as the owner of each file.

However, a file's owner is stored as a number—the user's UID—on UNIX-based computers, but because the ls command displays the owner as a name and not a number, the ls command must look up the actual user name associated with the file owner's UID. Because UNIX UIDs and user names are stored in Active Directory, this lookup request must be serviced by Active Directory. If a large number of files are displayed when the ls command is run, this creates a substantial amount of lookup traffic between the UNIX computer and the Active Directory domain controller.

Server Suite reduces this traffic by caching the lookups so that the information does not have to be retrieved from the Active Directory each time a lookup is required. Commands such as ls check the local cache first for the relevant information instead of retrieving the information from Active Directory every time.

Troubleshooting Logon Failures

If a user attempts to log on to a computer that is in a Server Suite zone and the logon fails, the problem is typically caused by one of the following:

- Users attempting to log on to a computer they are not authorized to use.
- Users have an incomplete profile in the zone where the computer they are attempting to use is located.
- Users have not been assigned an appropriate role that allows logon access.
- Users have typed their non-Active Directory password or typed the wrong password more times than allowed.
- Local or group policy settings are applied to the computer to prevent access.

To investigate these potential problem areas:

1. Check whether the local UNIX computer can connect to the Active Directory domain controller.
    - Log on to the computer using a locally authenticated user, such as the local root user.
    - Run the ping command with the name of an appropriate domain controller in the forest.

      For example, if the local computer is joined to the snowline.org forest, the command might look similar to this:

      ```
      su -
      Password:
      ping shasta.snowline.org
      ```

      If the command receives a reply from the domain controller, the DNS service is functioning and the local computer is able to locate the domain controller on the network. If the ping command does not generate a reply, you should check your DNS configuration and check whether the local computer or the domain controller is disconnected from the network.
2. Check Active Directory information by running the adinfo command. The output from this command should appear similar to the following:

   ```
   Local host name: magnolia
   Joined to domain: snowline.org
   Joined as: magnolia.snowline.org
   Current DC: shasta.snowline.org
   Preferred site: Default-First-Site-Name
   Zone: snowline.org/Acme/Zones/cascade
   Last password set: 2017-12-21 11:37:22 PST
   CentrifyDC mode: connected
   ```

   If the mode is disconnected, check whether adclient is running and network connectivity. On a slow network adclient may drop the connection to Active Directory if there is a long delay in response time.

   If the output displays an <unavailable> error, you should try running the adleave command to leave Active Directory, re-run the adjoin command, then re-run the adinfo command. For example:

   ```
   adleave --force
   adjoin --user skip --zone cascade snowline.org
   Password:
   adinfo
   ```

   If a problem still exists, check the DNS host name of the local computer and the domain controller, the user name joining the domain, and the domain name you are using.
3. Check the clock synchronization between the local UNIX computer and the Active Directory domain controller.

If the clocks are not synchronized, reset the system clock on the UNIX computer using the date command.

4. Check for denied users and groups in the /etc/centrifydc/centrifydc.conf file or the Login Controls group policy. For example, open the centrifydc.conf file in a text editor, such as vi:

   vi /etc/centrifydc/centrifydc.conf

   - Search for the pam.deny.users line and make sure that the user who is trying to log on is not listed.
   - Search for the pam.deny.groups line and make sure that the user who is trying to log on is not a member of any group that is listed on this line.

5. Check the contents of the system log files or the centrifydc.log file after the user attempts to log on. You can use information in this file to help determine whether the issue is with the configuration of the software or with the user's account.

6. Check for conflicts between local user accounts and the user profiles in Active Directory by running the getent command. For example:

   getent passwd

   This command displays a list of local and Active Directory users with access to the computer. If the user's name is not listed but other Active Directory users are listed, the problem may be in the user's Active Directory account settings or UNIX profile.

   If no Active Directory users are listed in the output of the command, check whether adclient is running and whether the Active Directory domain controller is available.

7. Check the user's Active Directory account settings using Access Manager or Active Directory Users and Computers. For example:

   - Check whether the user has a UNIX profile for the local computer's zone.
   - If the user has a UNIX profile in the zone, check whether the profile is currently enabled.
   - If the user has an enabled UNIX profile, check the user's group membership to determine whether it is a local group defined in a different domain than the computer account.
   - Check whether the user's account has been disabled or locked.
   - Check whether any user-based group policies have been applied to the user account that may prevent access to the UNIX computer.

8. Enable logging of adclient activity using the addebug command. For example:

   /usr/share/centrifydc/bin/addebug on

   This command enables extensive logging of each operation performed by the adclient process in the /var/log/centrifydc.log file. You can use the information in this file to further diagnose the cause of any problems or to enable Server Suite's support staff to assist with resolving any issues.

Evaluating Additional Services And Integrations

After you have deployed Server Suite Agents to implement an Active Directory-based security and directory services, you may want to explore other ways to take advantage of Active Directory's infrastructure. In evaluating ways to extend your security and directory services, you must first understand:

- How the UNIX servers and workstation that are joining the domain are used
- Which applications are accessed locally and which applications are accessed by remote users
- How the servers and workstations are managed, and whether administrators are local users or remote users
- Whether there are specific additional IT services you want to enable
- Whether there are specific controls you want to apply

As a starting point, you should consider whether computers joining the domain are workstations that primarily support local logon sessions or servers that require few, if any, local logon sessions. In many cases, UNIX computers have few interactive users but are frequently used as application servers that host database or web applications. For those computers, you should determine whether Active Directory authentication and authorization is primarily for administrators who manage the server or for users who log in to access the application.

Some of the ways you can extend and evolve the deployment of Server Suite software include:

- Adding authentication service for applications
- Adding custom reports for auditing UNIX properties
- Adding group policies
- Adding support for NIS clients
- Using programs optimized for Kerberos authentication
- Integrating with products from other vendors

Adding Authentication Service for Applications

Because Active Directory and Server Suite use Kerberos and LDAP standards, many Kerberos-enabled or PAM-enabled applications can use Active Directory for authentication and authorization service with little or no configuration. One way you can evolve your deployment is to add support for single sign-on to additional applications.

Supporting Single Sign-on for Kerberos-enabled Applications

The primary way that Server Suite supports single sign-on is through Kerberos. Kerberos provides a ticket-based authentication mechanism that is the default method for authentication in Active Directory. When a user logs on to a computer that uses Active Directory authentication, a Kerberos ticket is issued to the user and that ticket allows the user to access data, applications, other computers, and other sessions without having to present credentials again. This silent authentication that takes place in the background as users browse network shares or run applications is the key to enabling a single sign-on experience.

Many applications are Kerberos-enabled by default or can be configured to support the use of Kerberos tickets. By default, when a computer joins an Active Directory domain, Kerberos requests are forwarded and serviced by the Kerberos Key Distribution Server on the Active Directory domain controller. Therefore, in most cases, existing Kerberos-enabled applications can authenticate and authorize access without any modification.

If you use an application that is not configured to use Kerberos authentication by default, however, you may need to modify configuration options or use specific command line options to support single sign-on.

In addition, users must be assigned to a role with the **Non-password (SSO) login is allowed** system right. This right is enabled in the predefined UNIX Login role. If you create custom roles and want to allow single sign-on, you should select this system right when defining the role.

Supporting Single Sign-on for PAM-aware Applications

Pluggable Authentication Modules (PAM) provide a flexible mechanism for authenticating users regardless of the underlying authentication system. Most programs and applications that rely on user authentication use PAM.

The agent uses its own PAM module, pam_centrifydc.so, to direct PAM requests to Active Directory. Therefore, in most cases, existing PAM-enabled applications can authenticate and authorize access without any modification and support single sign-on without any special configuration.

One known exception, however, is that most database applications support PAM authentication, but do not enable it by default. To support single sign-on for database applications, you should modify the database configuration to enable PAM authentication.

In addition, users must be assigned to a role with the **Non-password (SSO) login is allowed** system right. This right is enabled in the predefined UNIX Login role. If you create custom roles and want to allow single sign-on, you should select this system right when defining the role.

Supporting Active Directory Authentication for Apache and Java Applications

Most Web and J2EE platforms provide their own native authentication and authorization services for Web developers to use. With Server Suite, you can choose to extend the native interfaces to enable web applications to provide single sign-on capability or redirect authentication requests to Active Directory instead of a native authenticator.

Supporting Database Server Applications

Most database servers provide their own native authentication and authorization services. With Server Suite, you can choose to extend the native interfaces to enable supported database servers to provide single sign-on capability or redirect authentication requests to Active Directory instead of a native authentication service.

Adding Custom Reports for Auditing Unix Properties

Server Suite includes several default reports that you can use to monitor and audit access to the computers in your environment. The default reports provide detailed information about your UNIX users, groups, computers, zones, and licenses, and enable you to verify which users have access to specific computers, zones, or applications. Default reports also provide easy access to the information that you require for auditing, business planning, and regulatory compliance. After you generate a report, you can save the report in the following formats:

- Microsoft Excel (.xls)
- Microsoft Word (.doc)

- Adobe Acrobat (.pdf)
- XML document (.xml)

For example, after generating a report with information about the users in each zone, you can save it as a Microsoft Excel spreadsheet (.xls), and import the information into an Excel Workbook to create a Charge Back report on account usage for each department.

One of the most common ways to evolve the Server Suite deployment is to create custom reports that are specifically tailored to your organization and auditing requirements. The Access Manager console includes a Report Wizard that allows you select the specific Active Directory objects and properties and the relationships on which you want to base the report.

For information about creating and generating custom reports, see the *Administrator's Guide for Linux and UNIX*.

Adding Group Policies

For many companies, centralized policy management and configuration control is just as important as centralized identity management. With Server Suite, you can apply group policy settings from Active Directory to the non-Windows computers and UNIX users.

Evaluating Existing Policy Settings

If you have applied any domain-wide policies in the Active Directory forest, you should review what the policy settings are and where they are enforced for Windows-based computers. You should then evaluate which policy settings, if any, are applicable for computers running UNIX, Linux, or Mac OS X operating systems. For example, most organizations establish a policy for password complexity. You can view your current password policy settings by clicking **Domain Security Policy** under **Administrative Tools** to open the **Default Domain Security Settings**, then select **Password Policy**.

If you enable any password policy settings for the domain, they automatically apply to UNIX users and managed computers because Active Directory uses these settings when authenticating users. If you enable or change any of the default domain policy settings, you should consider how they affect UNIX users and computers. For information about the standard Windows group policies that apply for UNIX, see the Group Policy Guide.

Adding Server Suite-specific Group Policies to a GPO

You can add Server Suite-specific configuration settings to any Group Policy Object applied to any site, domain, or organizational unit in the Active Directory forest. You can then manage the specific policies enabled and settings applied centrally through the Group Policy Object Editor on Windows.

Each GPO can consist of configuration information that applies to computers, configuration information that applies to users, or sections of policy that apply specifically either to users or to computers. You link a GPO to an Active Directory organizational unit, domain, or site. Windows then applies the policy settings based on an established hierarchical order.

The Server Suite-specific group policy settings available for users and groups are defined separate administrative templates (.adm or .xml files) that can be added to any GPO. If you enable any of the policy settings, they are written to a virtual registry on the UNIX computer. The Server Suite Agent then runs a set of local mapping programs that read the virtual registry and modify local configuration files to implement the setting defined by the group policy. You can also create your own custom administrative template and mapper programs to implement custom group policies.

For more detailed information about creating and managing Server Suite-specific group policies, see the *Group Policy Guide* and Active Directory documentation.

Adding Support for NIS Clients

You can extend Server Suite software to provide NIS service from a Server Suite-managed computer, acting as a NIS server, to NIS client requests using Active Directory as the central data store for NIS maps.

There are many scenarios in which adding the Server Suite Network Information Service (adnisd) to your infrastructure can enable you to integrate Server Suite and Active Directory with other enterprise solutions. For example, the adnisd Network Information Service and Server Suite zones can be used to centrally manage and map multiple UNIX identities to a Windows user account for access resources stored on EMC Celerra Network Servers or Network Appliance Filers. Active Directory remains the central identity store and zones remain the primary way of mapping UNIX profiles to a user account, but the Server Suite Network Information Service enables you to deliver the appropriate information to servers and devices across the network.

Using the Server Suite Network Information Service in conjunction with the Server Suite Agent is a scenario like this provides the following advantages:

- **Redundancy.** As an NIS client, the Celerra Network Server can find NIS servers by broadcasting on the local subnet. If a subnet hosts more than one Server Suite-managed computer acting as a NIS server, the Celerra or NetApp server can fail over from one NIS server to another NIS server on that subnet, thus enabling multiple NIS paths to the same data held within Active Directory.

- **Multi-domain support.** The Server Suite NIS service can provide user mapping data to NIS clients who may have an account anywhere within an Active Directory forest, including remote or child domains. Through the Active Directory Global Catalog, Server Suite Agents can find user mapping information for users anywhere across the forest.

Extending your deployment with the Server Suite Network Information Service also enables you to centrally manage network information and publish custom information to NIS clients throughout the network without modifying the underlying Active Directory schema.

Using Programs Optimized for Kerberos Authentication

As a management platform, Server Suite provides its own versions of commonly-used open source programs. For example, the following packages have been optimized to work with Server Suite software and Active Directory:

- Standard MIT Kerberos utilities, such as kinit and kdestroy, are installed with the agent to support Kerberos keytab management for accounts in Active Directory.
- Kerberos-enabled client programs such as OpenSSH, support Kerberos authentication and single sign-on for secure connections between Server Suite-managed computers.
- An enhanced version of PuTTY supports Kerberos authentication for secure, remote access from Windows computers to Server Suite-managed computers.

Integrating with Products from Other Vendors

Server Suite software also integrates with products from many other vendors, such as Splunk, IBM DB2, SAP Netweaver and Secure Network Communication (SNC), and Quest ActiveRoles Server. In addition to Active Directory, you can use Server Suite software to extend other Microsoft services such as Services for Network File System (NFS), Microsoft Identity Integration Server (MIIS), and Microsoft Active Directory Federation Services (ADFS).

For more information about add-on packages, integration with other systems, or configuring Server Suite software to work with other products, see the Resource Center on the Delinea website.

Getting Assistance from Support

It is recommended that you take the following steps if you need assistance with an issue or have questions about the operation of Server Suite components:

1. Check the Support Portal on the Delinea website to search the Knowledge Base to see if your problem is a known issue or something for which there is a recommended solution.
    - Open http://www.centrify.com/support/login.asp in a Web browser.
    - Log in using your customer account information and password.
    - Click **Find Answer** and type one or more key words to describe the issue, then click **Find** to view potential answers to your question. For example, to search for known issues, type known issues and click **Find** to see articles related to the known issues in different releases.

      If your issue is not covered in an existing Knowledge Base article or the Server Suite documentation set, you should open a case with Delinea Support.
2. Click **Log a Case** to open a new case using the Delinea Support Portal.

   Alternatively, you can contact Delinea Support by email or telephone, if you prefer. Worldwide contact information is available in the "How to open a case and collect information for Delinea Support" Knowledge Base article (KB-0301).
3. Provide as much information as possible about your case, including the operating environment where you encountered the issue, and the version of the Delinea product you are working with, then click **Submit** to open the case.

Before or after opening a support case, you may need to collect additional information about your environment. To help ensure your issue gets resolved quickly and efficiently, you should take the following steps to gather as much information as possible:

1. Verify the Server Suite Agent is running on the computer where you have encountered a problem. For example, run the following command:

   ps –ef | grep adclient

   If the adclient process is not running, check whether the watchdog process, cdcwatch, is running:

   ps –ef | grep cdcwatch

The cdcwatch process is used to restart adclient if it stops unexpectedly.

2. Enable logging for the Server Suite Agent. For example:

   /usr/share/centrifydc/bin/addebug on

3. Create a log file for the Mac OS X Directory Service. For example:

   killall –USR1 DirectoryService

4. Run the adinfo command to generate a report that describes the domain and current environment. For example:

   adinfo --diag --output filename

5. Duplicate the steps that led to the problem you want to report. For example, if an Active Directory user can't log in to a Server Suite managed computer, attempt to log the user in and confirm that the attempt fails. Be sure to make note of key information such as the user name or group name being, so that Delinea Support can identify problem accounts more quickly.

6. Verify that log file /var/log/centrifydc.log or /var/adm/syslog/centrifydc.log exists and contains data.

7. Generate information about Active Directory domain connectivity and configuration files by running the following command:

   adinfo --support

   This command writes output to the file /var/centrify/tmp/adinfo_support.txt.

8. If there is a core dump during or related to the problem, save the core file and inform Delinea Support that it exists. Delinea Support may ask for the file to be uploaded for their review.

   If the core dump is caused by an agent process or command, such as adclient or adinfo, open the /etc/centrifydc/centrifydc.conf file and change the adclient.dumpcore parameter from never to always and restart the agent:

   /etc/init.d/centrifydc restart

9. If there is a cache-related issue, Delinea Support may want the contents of the /var/centrifydc directory. You should be able to create an archive of the directory, if needed.

10. If there is a DNS, LDAP, or other network issue, Delinea Support may require a network trace. You can use Ethereal to create the network trace from Windows or UNIX. You can also use Netmon on Windows computers.

11. Create an archive (for example, a .tar or .zip file) that contains all of the log files and diagnostic reports you have generated, and add the archive to your case or send it directly to Delinea Support.

12. Consult with Delinea Support to determine whether to turn off debug logging. If no more information is needed, run the following command:

    /usr/share/centrifydc/bin/addebug off

Templates and Sample Forms

This section provides templates and samples that you can customize and use in the deployment process. These templates represent documents that are commonly used, such as change control requests and email notifications of software changes. Your organization may require you to use organization-specific versions of these documents.

Simplified Environment Analysis and Zone Design Template

This template provides a framework for the information that the deployment team should collect, analyze, and document in evaluating the existing network infrastructure and how it will change after deployment. Depending on your environment and requirements, you might need to collect additional information, but this template describes the most common elements with examples that you can adapt to your organization.

1. Introduction

   Use this section to provide a brief overview of the deployment plan. For example, document the features you plan to deploy, any primary goals that might affect design decisions, and any dependencies or special considerations, such as activities that require change control approval or enhanced permissions.

2. Network architecture

   Use this section to capture details about your existing network configuration and Active Directory architecture. For example, you might want to record information about the Active Directory site, forest, and domain controllers, including trust relationships and domain and forest functional levels, if applicable.

   You might also include details about your DNS configuration, including whether you have more than one DNS namespace and any port requirements, firewall restrictions, and any network connectivity issues. For details about the default ports used, see Default ports for network traffic and communication.

3. Server Suite-managed computers

   Use this section to provide details about the existing UNIX, Linux, and Mac OS X computers on which you plan to deploy the Server Suite Agent.

4. Provisioning process

   Use this section to describe the process for provisioning computers, groups, and users.

5. Rights, roles, and role assignments

   Use this section to describe the rights, roles, role assignments, and configuration policies you require. For example, if you use the sudo program and the sudoers file, use this section to document how rights and roles defined in the sudoers file and whether the sudoers file is managed locally on each computer or in a central location.

6. Zone architecture

   Use this section to identify the Active Directory schema you are using and where Server Suite-related objects are located in the Active Directory forest.

7. Deployment preparation in Active Directory

   Use this section to summarize the deployment of Server Suite components into the existing Active Directory forest and domain.

8. Windows installation

   Use this section to describe how zones will be created and configured.

9. UNIX deployment

   Use this section to describes the deployment of Server Suite Agents on UNIX computers.

10. Group Policies

    Use this section describes the group policies that will be deployed for UNIX computers.

Change Control Request Form

Delinea

Most larger organizations require a formal change request to be submitted for any changes to Active Directory. The purpose of this template is to illustrate a request for creating new Active Directory organizational units, groups, and users. If the deployment team is not allowed to add UNIX groups and group members to Active Directory after the organizational structure if created, it is likely the project will experience delays.

Computer:

Change Requested:

Approved By:

Test Case Matrix Sample

To validate the pilot deployment, most organizations execute at least some formal testing of features and functionality. The purpose of this template is to suggest a basic set of test cases to execute that apply to most environments. These test activities apply to setting up your environment, installing the software, and performing common administrative tasks. You can skip any activities that don't apply to your organization.

Testing Matrix

Create the OU Structure with a script or manually	Active Directory setup activities	
Create the OU Permissions with a script or manually		
Create Security Groups with a script or manually		
Create Distribution Groups with a script or manually		
Create the Zones Container with the Setup Wizard, a script, or manually		
Create the Licenses Container with the Setup Wizard, a script, or manually		
Create the service account for the Zone Provisioning Agent		
Update the local or domain policy to allow the Zone Provisioning Agent service to Log on as a service right		
Deploy the agent on computers		
Create a zone with a script or Access Manager console	Access Manager console activity	
Delegate zone control with a script or using the Delegate Zone Control Wizard		
Pre-Create Computer account		
Import UNIX groups from group files or group NIS maps		
Resolve mapping issues		
Import UNIX users from passwd files or passwd NIS maps		
Assign interactive users to the UNIX Login role	Authorization activities	
Assign users who need profile but not access to the listed role		
Join computers to the domain using adjoin	You should prepare for migration and create one or more initial zones before you join the domain.	

Configure root-equivalent rights		
Configure root-equivalent replacement role		
Add an Active Directory group for the role		
Test role access		
Test role privileges control		
Identify current management process (manual or automated)	UID consolidation activities	
Document the new management process		
Define the business rule for assigning UIDs (for example, SID)		
Identify active users to preserve, migrate, and keep		
Run adfixid to change file ownership		
Identify current management process (manual or automated)	GID consolidation activities	
Document the new management process		
Define the business rule for assigning the primary GID values (for example, GID)		
Identify the Active Directory groups for primary GID assignments	Domain Users	
Validate Active Directory log on credentials	User login activities	
Validate successful access to UNIX, Linux, Mac OS X		
Validate successful application usage		
Validate password complexity policy		
Validate account lockout policy		
Validate role enforcement		
Validate single sign on		
Validate password reset		
Test period users validated	Clean up activities	
Test period groups validated		
Test period roles validated		
Run adrmlocal to remove local accounts		

Preliminary Software Delivery Notification Email Template

The purpose of this template is to notify users that they are scheduled to receive new software that will be delivered to their computers. This email notice

should include a specific delivery date or a time frame estimate, if possible. Although you can delete this information from the email message you send out in your organization, this notice is most effective if users know specifically when the change is scheduled to occur. You can also customize the specific requirements or objectives that Server Suite is helping your organization achieve.

Colleagues:

The *[Department Testing Server Suite]* has successfully completed testing of the Server Suite software and is ready to begin the deployment portion of the project. The target date for deployment is *[Scheduled time]*.

Deployment of this software will greatly enhance our ability to comply with multiple industry requirements to include *[List objectives, such as: PCI, Sarbanes-Oxley compliance, Internal/External Security Audit, specific organization initiatives]*. These requirements are in alignment with prioritized corporate business objectives.

The Server Suite software enables the streamlining of authentication, access controls and privileges, and auditing for all corporate IT systems. For the most part, deployment and streamlined authentication and authorizations services occurs "behind the scenes" with minimal, if any, user disruption. You should not notice any operational changes when the software is deployed to your computer.

Thank you for your cooperation,

[IT Department Signature]

Department-specific Announcement and Instructions Email Template

The purpose of this template is to notify users in a specific department that they are scheduled to transition to using Server Suite for authentication and authorization. This email notification indicates that you plan to join the computers in the department to an Active Directory domain during down time. Depending on your organization's policies, this email may suggest users log on with their Active Directory credentials or explicitly state that they can continue to log on with their existing credentials.

Colleagues:

The *[Specific department you are deploying to, such as: Accounting Department]* is scheduled to begin the transition to Server Suite next week. In order to ensure a smooth transaction we simply ask that you log off of all systems before leaving for the weekend. When you return to work the following week, you should *[be able to log on with your current user name and password]*.

If you experience any difficulties logging on, or with application connectivity, please submit a ticket or contact the support desk immediately. Several members of each department helped the IT team perform successful testing and validation of this new solution, and we anticipate a smooth transition.

Thank you for your cooperation,

[IT Department Signature]

General Announcement and Deployment Schedule Email Template

The purpose of this template is to notify a broader user community of the deployment schedule for multiple departments across the company. This sample also illustrates the type of notes you can incorporate into the email message to keep other groups informed of their status. The general announcement may also include portions of the other two email templates. For example, you may want to include the objectives the transition to Server Suite helps the company achieve or the instructions to use current or Active Directory credentials after migration.

Colleagues:

At the completion of the week, the *[Server Suite Deployment Project Team]* will allocate first response resources to the next department scheduled for deployment.

This is the schedule coordinated with the Department Heads throughout the company:

9 May 2017	Information Technology	
16 May 2017	Accounting	
23 May 2017	Marketing	Pending EOQ Reports

30 May 2017	Security	
6 Jun 2017	Sales	
13 Jun 2017	Executive	
20 Jun 2017	PMO	
27 Jun 2017	Data Warehouse	Pending EOQ Reports
3 Jul 2017	Training	
10 Jul 2017	Business Development	
17 Jul 2017	Audit	

The IT Department would like to thank everyone to date for their work on this project, and look forward to a successful deployment. If you have any questions, please submit them to the *[Server Suite_project]* distribution list and include your contact information. We will respond with answers or contact you directly for more information.

Sincerely,

[IT Department I Server Suite Deployment Project Team]

Deployment Team Task Checklist

Before you install the pilot deployment, you should prepare a deployment checklist to ensure you have the information you need to successfully complete the deployment. For example, you should review port requirements, verify DNS resolution, and create one or more spreadsheets that describe the user and group accounts to be imported and any special relationships, such as membership in specific groups that need to be preserved or any special configuration you want to implement.

Creating a deployment checklist is optional, but can help you to collect detailed information about each of the computers targeted for deployment.

The following example illustrates information you can collect and record in a deployment team task checklist.

	Operating system, version, and patch level for target computers
	Host name and IP address for target computers
	Current disk space for target computers
	Review the details of the current DNS configuration For example: Is the address resolved through a UNIX DNS server, Windows DNS server, or settings in the /etc/hosts and /etc/resolv.conf files? Is the computer using a DNS server that has SRV records for Active Directory domain controllers? Are UNIX subnets registered and associated with Sites in Active Directory? Are you using a disjointed DNS namespace, where a UNIX computer name may be server.company.com but the Active Directory domain name is server.windows.company.com? Are you using DNS aliases and do they resolve correctly? Are there multiple network interfaces (NIC) in use?
	Current network time provider (NTP) For example, does the computer use a different server to determine the time than the Active Directory domain controller?
	Current firewall configuration For example, are there any firewalls blocking required ports between the UNIX computer and the Active

	Directory domain controllers for the registered sites?
	Current applications and services For example, do you have Perl, Samba, or OpenSSH deployed? Are the versions you have compatible with the Server Suite Agent or—if a Server Suite version is available—to be replaced by versions provided by Server Suite? Do you have existing authentication providers deployed? Are existing applications and services Kerberos-enabled or PAM-enabled? Are there other applications that require local users or groups?
	Current source of user and group information For example, are the /etc/passwd and /etc/group files the only source of user information for the users who access this computer or other identity stores, such as existing LDAP servers or NIS domains, used? Are there any specific users or groups that should remain locally defined?
	Current NSS configuration For example, have you reviewed the contents of the nsswitch.conf file to check for other sources of user and group information?
	Connectivity between this computer and the domain controller For example, is there a reply from the domain controller when you run the ping command?
	User names and UIDs checked for conflicts across the target group
	Zone requirements analyzed for the target group
	Zone identified for this computer
	Server Suite Agent installed and the computer joined to the domain
	Groups allowed or denied access identified for this computer
	Existing users and groups for this computer imported into Active Directory
	Imported user and group profiles mapped to Active Directory accounts
	Allowed or denied groups configured using parameter values or group policy

If you use a deployment checklist, you can also include additional notes and details about the activities performed. For example, a partially completed checklist might look something like this:

	Operating system: Sun Solaris 10 with all patches applied (17-April-2017)
	Host name and IP address: aspen, 177.29.10.10
	Current DNS configuration: Resolved through the enterprise DNS server, spider.ajax.org
	Current time source is NTP server: ntpd on solstice.ajax.org Change for deployment: Use SNTP on the Active Directory domain controller
	Current firewall configuration: No port issues
	Existing OpenSSH version to be replaced, no other issues found.

	Current source of user and group information: /etc/passwd, /etc/group, and NIS domain nwest03 have users who access aspen
	Connectivity with the domain controller: Verified by JR (2-May-2011).
	User names and UIDs checked for conflicts across the target group: Analyzed by JR and DC (4-May-2017).
	Zone requirements analyzed for the target group: Zones required for the target group are nwest01, swest02, corp-main, and nwest03 (9 May 2017). SF to recommend new extended zone descriptions for approval.
	Zone identified for this computer: nwest03
	Server Suite Agent installed and the computer joined to the Active Directory domain: dc3colorado.ajax.org, OU: US-UNIX-Computers
	Groups allowed r denied access identified for this computer: Allowed access group—all_employees, oracle_sys Denied access—consultants, temps
	Existing users and groups for this computer imported into Active Directory: Completed by DC (20-May-2017).
	Imported user and group profiles mapped to Active Directory accounts: Work complete for users and groups that already had matching Active Directory candidates. Work in progress for the remaining profiles without any matching Active Directory candidate. Target date for completion: 31-May-2017
	Allowed or denied groups configured using parameter values or group policy: TBD

Permissions Required for Administrative Tasks

This chapter describes the permissions required to perform administrative tasks that affect Centrify-specific objects in Active Directory. You can set these permissions manually for individual users and groups who manage Centrify zones in Active Directory. However, setting permissions manually can be time-consuming and error-prone. In most cases, you should use the Zone Delegation Wizard to authorize users to perform specific tasks.

At a minimum, all Access Manager actions require users to have generic Read permission. This permission is typically granted to all Authenticated Users by default.

Because Authenticated Users have read access, they can run reports in the Report Center. No additional rights need to be granted to enable users to run reports.

How Permissions Are Set

Access Manager requires specific rights for administrators to work with objects such as UNIX users, groups, and computers within Active Directory. As part of your deployment planning process, you should review the rights required to set up and manage Centrifyspecific objects and be familiar with how to manually assign rights for managing Centrify objects, if needed.

Built-in Windows groups, such as Domain Admins and Domain Users, have default permissions, which might be customized for your organization. In general, the administrators for the forest root domain have broad authority to set permissions for all other users and groups, including the administrators of other domains. Therefore, whether you can modify the permissions for specific users and groups within your Active Directory environment will depend on the policies of your organization.

If you have the appropriate authority, there are several ways you can access, verify, and modify the permissions assigned to specific users and groups.

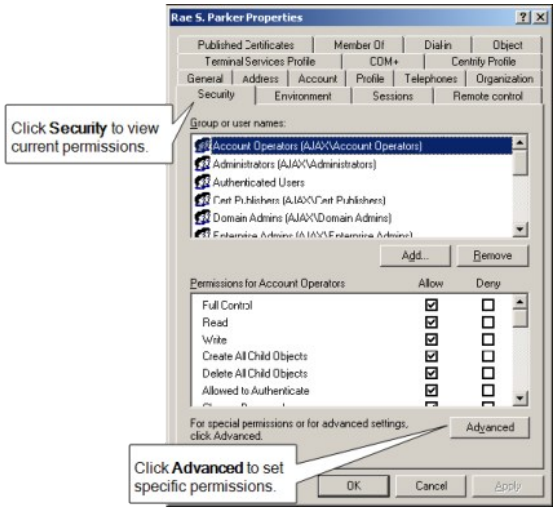
For example, you can view and modify permissions in the following ways:

- Use **ADSI Edit** to directly modify any Active Directory attributes.
- Use **Active Directory Users and Computers** to set basic or advanced permissions on any Active Directory object through the **Security** tab.

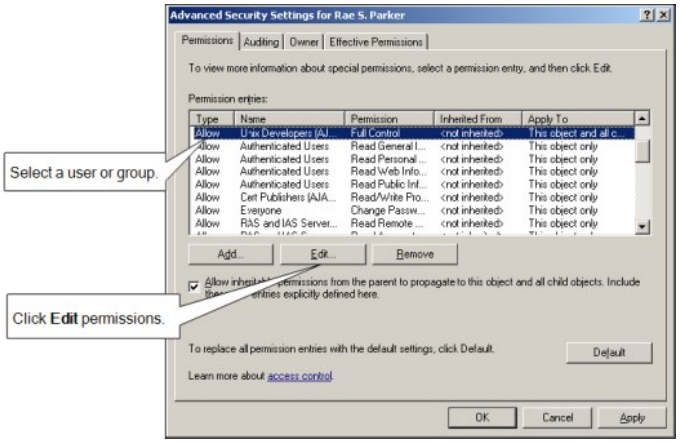
  To display the Security tab, select **View > Advanced Features**. To access some permissions, however, your user account must have **Create all child objects** or **Write all properties** permissions.
- Run the **Zone Delegation Wizard** to set the appropriate permissions for specific users or groups to perform specific tasks within a zone.
- Click **Permissions** when viewing Zone Properties in Access Manager to set basic or advanced permissions on any zone object.
- Click **Permissions** when viewing the Centrify Profile for a user in Access Manager to set basic or advanced permissions on any user object.

The following steps illustrate how you can set permissions from Active Directory Users and Computers:

1. Open the console and connect to the Active Directory domain.
2. Select an Active Directory object, such as a user or computer, rightclick, then click **Properties**.
3. Click the **Security** tab, then click **Advanced**.



4. Select the user or group to which you want to assign rights, then click **Edit**.

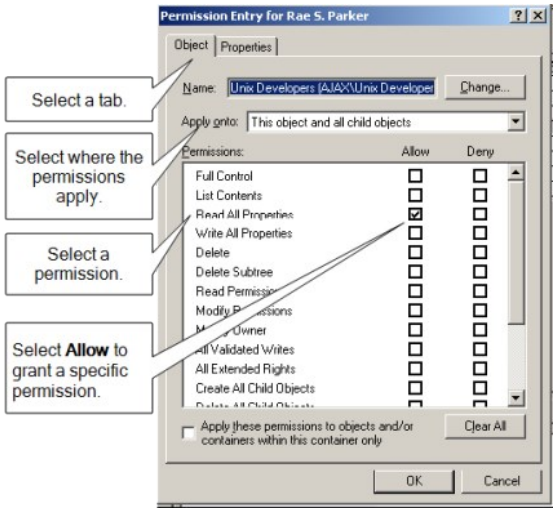


If the user or group to which you want to assign permissions isn't listed, click **Add** to find the account.

5. In the Permission Entry dialog box, click the **Object** or **Properties** tab, as needed.

   Selecting Object or Properties and where the permission should be applied varies depending on the task you are allowing a user or group to perform.

6. Select the specific rights you want to assign by scrolling to find the permission, then clicking the **Allow** check box.



7. When you are finished setting the appropriate permissions, click **OK**.

   For more specific information about how to set permissions on Active Directory objects and properties and how to view, modify, and remove permissions, see your Active Directory documentation.

Permissions Required to Use the Setup Wizard

In most cases, you run the Setup Wizard to guide you through the configuration of Active Directory for Centrify. The Setup Wizard updates Active Directory with Centrify-specific objects and properties, including zone and license containers that are required for proper operation.

To successfully perform initialization tasks, the user account that runs the Setup Wizard must have specific rights. Because some of these rights might be reserved for administrative accounts, some users might be prevented from performing certain steps in the Setup Wizard.

To allow other user accounts to run the Setup Wizard, you can manually create the appropriate container objects, then assign to those objects only the specific permissions needed to correctly complete the configuration of Centrify-specific objects. Users can then use the Setup Wizard to select the appropriate container objects and perform all of the necessary steps without being members of an administrative group.

Licenses Container Permission Requirements

The following table describes the minimum rights that must be applied to the Centrifyspecific container objects or other users to successfully complete the configuration of Centrify software.

Licenses container	Read all properties Create classStore Objects Modify permissions	This object only
	Write Description property Write displayName property	This object and all child objects
	The Setup Wizard requires you to create or select at least one parent container for license keys. By default, this container object is: domain/Program Data/Centrify/Licenses You can create additional License containers, if needed, through the Manage Licenses dialog box. By default, all Authenticated Users have read and list contents permission for the Licenses container and all of its child objects. You can change these permissions if you want to restrict access to Access Manager.	

Zones container or any container used as a destination for a new zone	Read all properties Create classStore Objects Create container objects	This object only
	Write displayName property	This object and all child objects
	The Setup Wizard requires you to create or select a parent container object for creating new zones. By default, this container object is: domain/Program Data/Centrify/Zones You can use other containers for zones, if needed. For example, if you have created a separate high-level organizational unit called UNIX as the parent container: domain/UNIX/Zones	
ZoneName/Computers container	Create group objects Write Description property	This object only
	These permissions are only needed if you are supporting “agentless” authentication in a zone.	
Computers container For example, the generic Computers container: domain.com/Computers	Write operatingSystem property Write operatingSystemVersion property Write operatingSystemHotfix property Write operatingSystemServicePack property	SELF on Computer objects
	These permission are granted to each computer’s SELF account when you select the Grant computer accounts in the Computers container permission to update their own account information option in the Setup Wizard.	

Licenses Container Permissions

The following table describes the minimum rights that must be applied to the Centrify Licenses container that stores the license keys for your installation.

Licenses container	Read all properties Create classStore Objects Modify permissions	This object only
	Write Description property Write displayName property	This object and all child objects

The Setup Wizard requires you to create or select a parent container for license keys. The default location for the parent container for license keys depends on the organizational structure you deploy. For example, if you use the recommended organizational structure, the default location for licenses would be domain/Centrify/Licenses.

You must have at least one parent container for license keys in the forest. You might want to create more than one license container objects to give you more granular control over who has access to which licenses.

By default, all Authenticated Users have Read and List Contents permission for the Licenses container and all of its child objects. These permissions are required to use Access Manager. You can change who has these permissions if you want to prevent users from using Access Manager.

Zones Container Permissions

The following table describes the minimum rights that must be applied to the Centrify Zones container.

Zones container or any container used as a destination for a new zone	Read all properties Create classStore Objects Create container objects	This object only
	Write displayName property	This object and all child objects
Change the default zone container	Delete	Previous zone container

The Setup Wizard requires you to create or select a parent Zones container object for new zones. The default location for the parent container for new zones depends on the organizational structure you deploy. For example, if you use the recommended organizational structure, the default location for new zones would be domain/Centrify/Zones. You can use other containers for zones or create multiple parent containers for zones to separate administrative duties for different groups.

Computers Container Permissions

The following table describes the minimum rights that must be applied to the generic Computers container (domain/Computers).

Computers container	Write operatingSystem property Write operatingSystemVersion property Write operatingSystemHotfix property Write operatingSystemServicePack property	SELF on Computer objects

These permission are granted to each computer's SELF account when you select the **Grant computer accounts in the Computers container permission to update their own account information** option in the Setup Wizard.

Computers Container Within a Zone Permissions

The following table describes the minimum rights that must be applied to the Computers container in a named zone if you are supporting "agentless" authentication in that zone.

ZoneName/Computers container	Create group objects Write Description property	This object only

Creating Parent Containers Manually

Some organizations prefer to create and manage Active Directory objects manually to ensure tight control over the objects and their related attributes. For example, you might want to manually create separate Zones or Licenses parent containers for different business units or geographic locations so that you can manually set different sets of permissions and related properties on those containers. Creating objects manually also enables you to have precise control over who has access to the objects.

You can create the container objects anywhere in the forest's directory structure, but you must have at least one parent Zones container object and at least one parent Licenses container object.

Optional Administrative Tasks

By default, Centrify does not require you to be an enterprise administrator or domain administrator of the forest root domain to install or configure Centrify-specific properties. However, some optional configuration tasks do require you to be an enterprise administrator or a domain administrator of the forest root domain.

These optional tasks involve:

- Creating display specifiers for Centrify profiles to enable access to the Centrify Profile properties page in Active Directory Users and Computers.
- Registering the administrative notification handler to ensure data consistency if users delete Centrify objects using Active Directory Users and Computers.

- Setting permissions for zones objects to enable maximum control over the placement of and rights associated with Centrify-related objects within Active Directory.

In most cases, if you want to perform any of these tasks, you must use an account that is an enterprise administrator or a domain administrator of the forest root domain.

Creating Display Specifiers for Centrify Profiles

To display the Centrify Profile properties in Active Directory Users and Computers, you must be an enterprise administrator or a domain administrator for the forest root domain because adding the Centrify Profile to Active Directory Users and Computers requires you to add display specifiers to Active Directory.

> **Note**: A display specifier is an Active Directory object that allows you to add components to the Active Directory Users and Computers (ADUC) Microsoft management console (MMC) snap-in.

If you want to make the Centrify Profile available in Active Directory Users and Computers, an enterprise administrator can manually define the display specifiers (under domain/Configuration/DisplaySpecifiers/LanguageID/) for computer, group, and user properties by modifying the adminPropertyPages attribute with the appropriate GUID. For example, if the Active Directory domain is ajax.org and the language you support is US-English (CN=409), you would define the display specifiers in:

ajax.org/Configuration/DisplaySpecifiers/409

> **Note**: Adding the display specifiers for Centrify properties is an optional step you can perform manually using ADSI Edit or by running the displayspecifier.vbs script. If you manage all Centrify objects through Access Manager, you do not need to perform this task.

To use the displayspecifier.vbs script to set up the display specifiers:

1. Log on using an enterprise administrator account or a domain administrator for the forest root domain.
2. Open a Command Prompt window and change to the Centrify installation directory. For example:

   cd C:\Program Files\Centrify\Access Manager
3. Run the displayspecifier.vbs script.

   If you want to manually add the display specifiers to display property pages in Active Directory Users and Computers, you must create the following entries using ADSI Edit, where n is the next number in the index of values for the attribute:

computer-Display displaySpecifier	adminPropertyPages	n,
group-Display displaySpecifier	adminPropertyPages	n, {0CDC9AD0-E870-483f-8D16-17EAB3B7F881}
user-Display displaySpecifier	adminPropertyPages	n, {543DBFE3-317D-4493-8D00-84591E4EDCDE}
inetOrgPerson-Display	adminPropertyPages	n, {543DBFE3-317D-4493-8D00-84591E4EDCDE}

For example, if the Active Directory domain is ajax.org and the language you support is US-English (CN=409), you would add these entries to the objects in:

ajax.org/Configuration/DisplaySpecifiers/409

In most cases, you only need to set up the display specifiers once for the Active Directory forest. If you support multiple languages, you can manually add the display specifiers to each language you support. For example, if your organization supports US-English (CN=409), Standard French (CN=40C), and Japanese (CN=411), you would add the display specifiers to these three containers. Once you have updated Active Directory by running the displayspecifier.vbs script or by manually adding the display specifiers, you can access the Centrify Profile properties using Active Directory Users and Computers.

Registering the Administrative Notification Handler

The administrative notification handler provides services to ensure data integrity in the Active Directory forest. You can register the notification handler automatically through the Setup Wizard the first time you start Access Manager, but this requires an account that is an enterprise administrator or a domain administrator in the forest root domain.

Registering the administrative notification handler is optional, but doing so helps to ensure that no orphan UNIX data is left in the directory if a user, group, or computer is deleted using Active Directory Users and Computers. When registered, the notification handler automatically deletes any service connection point (SCP) dependencies on a directory object if the directory object is deleted. Without this service, deleting a directory object such as a computer or user account in Active Directory might leave an orphan service connection point for the object in the directory.

If you don't want to perform this step in the Setup Wizard, you can manually configure the administrative notification handler using ADSI Edit or you can choose not to register the administrative notification handler for Centrify. If you choose not to register the administrative notification handler, however, you should periodically run the Analyze command to check for orphan data in the Active Directory forest.

To manually set up the administrative notification handler for Centrify, add the following entry using ADSI Edit under domain/Configuration/DisplaySpecifiers/LanguageID/ where n is the next number in the index of values for the attribute:

DS-UI-Default-Settings	dSUIAdminNotification	n,

For example, if the Active Directory domain is ajax.org and the language you support is US-English (CN=409), you would add this entry to the object in:

ajax.org/Configuration/DisplaySpecifiers/409

Granting Permissions For Administrative Tasks

The easiest way to grant permissions to perform administrative tasks is to use the Zone Delegation Wizard. The Zone Delegation Wizard enables you to delegate specific administrative tasks to specific users and groups. For each task you delegate to a specific user or group, you are providing that user or group with a specific set of permissions for working with objects in Active Directory.

The user who creates a zone has full control on the zone's serviceConnectionPoint. That user has exclusive permission to delegate administrative tasks to other users. The user who creates a zone is also the only user who can add NIS maps to the zone because creating NIS maps requires permission to create containers in Active Directory. The zone creator can, however, grant other users permission to add, remove, or modify NIS map entries.

The following table summarizes the permissions that can be assigned through your selections in the Zone Delegation Wizard. In addition to the permissions listed, however, the basic Read permission is required to perform any action. The Read permission is granted to Authenticated Users by default.

All	Permissions to perform all of the actions listed in the Zone Delegation Wizard and described below. This option allows a designated user or group to perform all of the other actions. Only the user who creates a zone can grant this permission to other users and groups for the zone.
Change zone properties	List contents on the ZoneName object container. Read all properties on the ZoneName object container. Write name on the ZoneName object container. Write Name on the ZoneName object container. Write Description property on the ZoneName object container.
Add users	List contents on the ZoneName/Users object container. Read all properties on the ZoneName/Users object container. Create serviceConnectionPoint objects on the ZoneName/Users object container.
Add groups	List contents on the ZoneName/Groups object container. Read all properties on the ZoneName/Groups object container. Create serviceConnectionPoint objects on the ZoneName/Groups object container.
Add local users	List contents on the ZoneName/Local Users object container. Read all properties on the ZoneName/Local Users object container. Add local users to the zone.
Add local groups	List contents on the ZoneName/Local Groups object container. Read all properties on the ZoneName/Local Groups object container. Add local groups to the zone.
Join computers to the zone	List contents on the ZoneName/Computers object container. Read all properties on the ZoneName/Computers object container. Create serviceConnectionPoint objects on the ZoneName/Computers object container. **Note** Joining the domain requires additional permissions on the Active Directory computer object, but the join command performs the necessary operations without requiring the additional permissions to be granted to the user or group you are designating as a trustee.

Remove zones	List contents on the ZoneName object container. Read all properties on the ZoneName object container. Allow Delete on the ZoneName object container. Allow Delete Subtree on the ZoneName object container.
Remove users	List contents on the ZoneName/Users object container. Read all properties on the ZoneName/Users object container. Delete serviceConnectionPoint objects on the ZoneName/Users object container.
Remove groups	List contents on the ZoneName/Groups object container. Read all properties on the ZoneName/Groups object container. Delete serviceConnectionPoint objects on the ZoneName/Groups object container.
Remove local users	List contents on the ZoneName/Local Users object container. Read all properties on the ZoneName/Local Users object container. Remove local users from the zone.
Remove local groups	List contents on the ZoneName/Local Groups object container. Read all properties on the ZoneName/Local Groups object container. Remove local groups from the zone.
Remove computers from the zone	List contents on the ZoneName/Computers object container. Read all properties on the ZoneName/Computers object container. Delete serviceConnectionPoint objects on the ZoneName/Computers object container.
Modify user profiles	List contents on the ZoneName/Users object container. Read all properties on the ZoneName/Users object container. Write cn on the serviceConnectionPoint object. Write name on the serviceConnectionPoint object. Write Name on the serviceConnectionPoint object. Write keywords on the serviceConnectionPoint object. For RFC 2307-compliant zones, modifying the user's UNIX profile also requires the following rights on the serviceConnectionPoint object of the UNIX user object: Write uid. Write uidNumber. Write loginShell. Write gidNumber. Write gecos. Write unixHomeDirectory. The additional rights for RFC 2307-compliant zones are applied to the posixAccount object associated with the serviceConnectionPoint for the UNIX user object.
Modify group profiles	List contents on the ZoneName/Groups object container. Read all properties on the object containers. Write name on the serviceConnectionPoint object. Write Name on the serviceConnectionPoint object. Write keywords on the serviceConnectionPoint object. For RFC 2307-compliant zones, modifying the group's UNIX profile also requires the following rights applied to the posixGroup object associated with the serviceConnectionPoint object of the UNIX group object: Write gidNumber.
Modify local user profiles	List contents on the ZoneName/Local Users object container. Read all properties on the ZoneName/Local Users object container. Modify local users in the zone. Parameters that can be modified are: User name (the UNIX login name). The user identifier (UID). The user's primary group profile numeric identifier (GID). The default home directory for the user. The default login shell for the user. General information about the user account (GECOS). State.
Modify local group profiles	List contents on the ZoneName/Local Groups object container. Read all properties on the object containers. Modify local groups in the zone. Parameters that can be modified are: Group name. Group members. Group identifier (GID). State.
Modify computer profiles	List contents on the ZoneName/Computers container object. Read all properties on the ZoneName/Computers container object. Write description on the ZoneName/Computers container object if the zone is a hierarchical zone. Write keywords on the serviceConnectionPoint object. Write displayName on the serviceConnectionPoint object. Write cn on the serviceConnectionPoint object. Write name on the serviceConnectionPoint object.
Allow computers to respond to NIS client requests	List contents on the ZoneName/Computers/zone_nis_servers group object. Read all properties on the ZoneName/Computers/zone_nis_servers group object. Write member property of group object on the ZoneName/Computers/zone_nis_servers group object.
	List contents on the ZoneName/Users and ZoneName/Groups container object. Read all properties on the ZoneName/Groups container object. Create serviceConnectionPoint on the ZoneName/Users and ZoneName/Groups container objects. Write cn on the

Import users and groups to zone	serviceConnectionPoint object. Write name on the serviceConnectionPoint object. Write managedby on the serviceConnectionPoint object. Write displayName on the serviceConnectionPoint object. Write keywords on the serviceConnectionPoint object. For RFC 2307-compliant zones, importing users also requires the following rights on the serviceConnectionPoint object of the UNIX user object ZoneName/Users: - Write uid. - Write uidNumber. - Write loginShell. - Write gidNumber. - Write unixHomeDirectory. - Write gecos. For RFC 2307-compliant zones, importing groups also requires the following right on the serviceConnectionPoint object of the UNIX group object under ZoneName/Groups: - Write gidNumber.
Manage roles and rights	List contents on the AzTask container and all child objects. Read all properties on the AzTask container and all child objects. Create msDS-AzTask objects Delete msDS-AzTask objects Write msDS-AzApplicationData on the msDs-AzTask object. Write cn on the msDs-AzTask object. Write name on the msDs-AzTask object. Write description on the msDs-AzTask object. Write msDs-OperationsForAzTask on the msDs-AzTask object. List contents on the AzOperation container and all child objects. Read all properties on the AzOperation container and all child objects. Create msDS-AzOperation objects Delete msDS-AzOperation objects Write msDs-AzApplicationData on the msDs-AzOperation object. Write cn on the msDs-AzOperation object. Write name on the msDs-AzOperation object. Write description on the msDs-AzOperation object. List contents on the msDS-AzAdminManager object. Read all properties on msDS-AzAdminManager object. Write msDs-AzApplicationData on msDS-AzAdminManager object.
Manage role assignments	List contents on the msDS-AzAdminManager object and all child objects. Read all properties on the msDS-AzAdminManager object and all child objects. Create msDS-AzRole objects. Delete msDS-AzRole objects. Write msDS-AzApplicationData on the msDS-AzRole object. Write msDS-TasksForAzRole on the msDS-AzRole object. Write msDS-MembersForAzRole on the msDS-AzRole object. Write displayName on the msDS-AzRole object. Write msDS-AzApplicationData on the msDS-AzAdminManager object.
Modify computer roles	List contents on the ZoneName object and all child objects. Read all properties on the ZoneName object and all child objects. Write msDS-AzApplicationData Write msDS-AzScopeName Write description
Add or remove NIS map entries	List contents on the ZoneName/NISMaps object container. Read all properties on the ZoneName/NISMaps object container. Create classStore Objects on the ZoneName/NISMaps object container. Write name on the ZoneName/NISMaps object container. Write Name on the ZoneName/NISMaps object container.
Modify NIS map entries	List contents on the ZoneName/NISMaps object container. Read all properties on the ZoneName/NISMaps object container. Write adminDescription on the classStore object. Write Description on the classStore object. Write wWWHomePage on the classStore object.
Remove NIS maps	List contents on the ZoneName/NISMaps object container. Read all properties on the ZoneName/NISMaps object container. Allow Delete on the MapName object. Allow Delete Subtree on the MapName object.

Note: In some cases, the permissions granted through the Zone Delegation Wizard are a subset of the complete permissions required to perform some tasks. For information about the complete permissions required to perform a specific task, see the section that describes the permissions for performing that task. For example, for information about setting permissions for NIS maps, see Setting permissions for NIS maps.

Setting permissions for zones

The user who creates a zone has full control over zone properties and administrative tasks. Only the zone owner can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. In most cases, the users who are allowed to create zones have domain administrator privileges and sufficient permissions to perform all administrative tasks and to delegate administrative tasks to other users.

If you manually set permissions to allow domain users to create zones, however, you should also manually set the permissions to allow those users to manage rights and roles or notify zone administrators that they should run the Zone Delegation Wizard and assign those tasks to their own account or to appropriate users and groups. At least one administrator must have permission to add an authorization store, define rights and roles, and manage role assignments in each zone. All users must have at least one valid role assignment to access a zone.

Creating a Zone

To create new zones, your user account must be set with the following permissions:

Parent container for new zones you created or selected in the Setup Wizard. For example: domain/UNIX/Zones	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: Create Container Objects Create Organizational Unit Objects **Note** Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create group objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Write Description property These permissions are only needed if you are supporting "agentless" authentication in the new zone.

Opening Zones

To open an existing zone, your user account must be set with the following permissions:

Parent container for new zones For example: domain/UNIX/Zones	On the **Object** tab, select **Allow** to apply the following permission to this object: List contents
Container for the individual zone For example, a ZoneName container object, such as: domain/UNIX/Zones/arcade	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributes Read allowedAttributesEffective Read canonicalName Read Description Read displayName Read name Read objectClass
Parent container for Users in the zone	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass
Parent container for Groups in the zone	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass

Modifying Zone Properties

To modify zone properties for a zone, your user account must be set with the following permissions:

Container for an individual zone For example, a ZoneName container object: domain/UNIX/Zones/arcade	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read Name Read name Read Description Read displayName Write Description **Note** You can grant these permission to specific users or groups by selecting the **Change zone properties** task in the Zone Delegation Wizard. These permissions also enable you to change the parent zone for a selected zone object.

Renaming a Zone

To rename a zone, your user account must be set with the following permissions:

Container for an individual zone For example, a ZoneName container object, such as: domain/UNIX/Zones/arcade	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Write name property Write Name property **Note** You can grant this permission to specific users or groups by selecting the **Change zone properties** task in the Zone Delegation Wizard.

Deleting a Zone

To delete a zone from Active Directory, your user account must be set with the following permissions:

Container for an individual zone For example, a ZoneName container object, such as: domain/UNIX/Zones/arcade	On the **Object** tab, select **Allow** to apply the following properties to this object only: Delete Delete Subtree Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read Name Read name Read displayName **Note** You can grant this permission to specific users or groups by selecting the **Delete zone** task in the Zone Delegation Wizard.

Managing Roles and Rights in a Zone

To manage rights and roles in a zone, including creating and deleting role definitions and updating time constraints, your user account must be set with the following permissions:

Container for the authorization store For example: domain/UNIX/Zones/arcade/Authorization	On the **Object** tab, select **Allow** to apply the following properties to this object and all child objects: List contents Read all properties Click the **Properties** tab and select **Allow** to apply the following properties to the msDS-AzAdminManager object: Write msDS-AzApplicationData
AzTaskObjectContainer	On the **Object** tab, select **Allow** to apply the following properties to this object and all child objects: List contents Read all properties Create msDS-AzTask objects Delete msDS-AzTask objects Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzTask objects: Write msDS-AzApplicationData Write cn Write name Write description Write msDs-OperationsForAzTask
AzOpObjectContainer	On the **Object** tab, select **Allow** to apply the following properties to this object and all child objects: List contents Read all properties Create msDS-AzOperation objects Delete msDS-AzOperation objects Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzOperation objects: Write msDS-AzApplicationData Write cn Write name Write description

Managing Role Assignments in a Zone

To manage role assignments in a zone, your user account must be set with the following permissions:

Container for the authorization store For example: domain/UNIX/Zones/arcade/Authorization	On the **Object** tab, select **Allow** to apply the following properties to this object only: List contents Read all properties Create all child objects Delete all child objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Write msDS-AzApplicationData Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzRole objects: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole
Computers container in the zone	On the **Object** tab, select **Allow** to apply the following properties to this object only: Create Container Right This permission is required to allow a delegated user to make the first role assignment after a computer is joined to Active Directory.
AzRoleObjectContainer	On the **Object** tab, select **Allow** to apply the following properties to the msDS-AzApplication object and all child objects: List contents Read all properties Create msDS-AzRole objects Delete msDS-AzRole objects Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzRole objects: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzAdminManager objects: Write msDS-AzApplicationData
AzOpObjectContainer	On the **Object** tab, select **Allow** to apply the following properties to this object only: Read all properties Create msDS-AzOperation objects Delete msDS-AzOperation objects Create msDS-AzRole objects Delete msDS-AzRole objects Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzRole objects: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole Click the **Properties** tab and select **Allow** to apply the following properties to msDS-

	AzOperation objects: Read name Read Name Write msDS-AzApplicationData Write name Write description

Changing Computer Role Properties in a Zone

To manage computer role properties in a zone, your user account must be set with the following permissions:

Container for the authorization store For example: domain/UNIX/Zones/arcade/Authorization/guid The *guid* object is a globally unique identifier (GUID) for the Authorization object. For example: CN=cab186af-61a0-4d54-a0dd...	On the **Object** tab, select **Allow** to apply the following properties to this object only: Read all properties Click the **Properties** tab and select **Allow** to apply the following properties to msDS-AzScope objects: Read name Read Name Write msDS-AzApplicationData Write msDS-AzScopeName Write description

Setting Permissions to Join or Leave the Domain

To join a UNIX computer to an Active Directory domain without predefining a computer account, your Active Directory user account must be set with the following permissions:

Parent container object for computer accounts For example: domain/UNIX/Servers	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects **Note** You can grant this permission to specific users or groups by selecting the **Join computers** task in the Zone Delegation Wizard.

To join a UNIX computer to an Active Directory domain and place the computer account in a specific organizational unit (OU), the Active Directory account used to join the domain must be set with the following permissions:

Parent container object for the computer accounts	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects Create Computer Objects

To join a UNIX computer to an Active Directory domain when you are using a predefined computer account, your Active Directory user account must be set with the following permissions:

Parent container object for the computer account	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects
Computer account object in Active Directory For example, if the computer account is AJAX in the default Active Directory Computers container: domain/Computers/AJAX	On the **Object** tab, select **Allow** to apply the following permission to this object only: Full Control This permission is required for enabling or disabling a computer account.

To remove a UNIX computer from an Active Directory domain, your Active Directory user account must be set with the following permissions:

Parent container object for the computer account	On the **Object** tab, select **Allow** to apply the following permission to this object only: Delete serviceConnectionPoint Objects If you are deleting a computer account, you also need the Delete Computer Objects permission.

Note: This setting only gives the user or group permission to leave an Active Directory domain. If you want to grant permission for a user or group

to delete a computer account, you also need the Delete Computer Objects permission.

Setting Permissions for Zone Computers

Although joining or leaving a domain is the primary task for working with computer accounts in Active Directory, there are also specific permissions required to list computers or modify computer properties. The objects and permissions can also vary depending on the type of zone the computer account is associated with and the task to be performed. In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard.

In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard instead of assigning the permissions manually.

Joining a Computer to a Zone

To join a computer to a zone, your user account must have the following permission:

Parent container object for the computer account in the zone For example, in a classic zone, the ZoneName/Computers container object: domain/UNIX/Zones/acme/Computers	Click the **Object** tab and select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects
Computer account object in Active Directory For example, if the computer account name is AJAX: domain/UNIX/Servers/AJAX	Click the **Object** tab and select **Allow** for the Full Control permission for the user with permission to join the domain. The adjoin command grants the computer's SELF account the following permissions: Write operatingSystem Write operatingSystemVersion Write operatingSystemHotfix Write operatingSystemServicePack Write servicePrincipalName Write userAccountControl Write dnsHostName

Listing Computer Accounts

To list computers, your user account must have the following permission:

Parent container object for the computer account in Active Directory For example: domain/UNIX/Servers	On the **Object** tab, select **Allow** to apply the following permission to this object for each of the computers to be included in the list: List contents
Parent container object for the computer account in the zone For example, in a classic zone, the ZoneName/Computers container object: domain/UNIX/Zones/acme/Computers	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass
The serviceConnectionPoint object for the computer account For example, if the computer account name is AJAX, select: domain/UNIX/Servers/AJAX then select: serviceConnectionPoint objects	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each of the computers to be included in the list: Read displayName Read keywords Read managedBy Read Name Read objectClass
Computer account object in Active Directory For example, if the computer account name is AJAX: domain/UNIX/Servers/AJAX	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each of the computers to be included in the list: Read objectClass Read Operating System Read Operating System Version Read userAccountControl

Modifying Computer Properties

To modify any computer account properties for a UNIX computer, your user account must have the following permission:

Parent container object for the computer account in Active Directory For example: domain/UNIX/Servers	On the **Object** tab, select **Allow** to apply the following permission to this object only: List contents
The serviceConnectionPoint object for the computer account For example, if the computer account name is AJAX, select: domain/UNIX/Servers/AJAX then select: serviceConnectionPoint objects	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributes Read allowedAttributesEffective Read displayName Read keywords Read managedBy Read Name Read objectClass Write keywords
Computer account object in Active Directory For example, if the computer account is AJAX in the default Active Directory Computers container: domain/UNIX/Servers/AJAX	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID Read objectSid Read objectClass Read Operating System Read Operating System Version Read userAccountControl

Responding to NIS Requests

If you are supporting "agentless" authentication or want to allow a computer to service NIS client requests in a zone, the computer must be a member of the zone_nis_servers group in the zone. Setting or unsetting the **Allow this computer to respond to NIS client requests** property requires the following permissions:

The zone_nis_servers group object For example, select: domain/UNIX/Zones/acme/Computers/zone_nis_servers	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: List contents Read all properties Write member property If the zone_nis_servers group does not already exist in the current zone, setting the **Allow this computer to respond to NIS client requests** property also requires the following permission on the ZoneName/Computers object: Create group objects

Changing the Computer Zone

If you need to change the zone for a computer account, your user account must have the following additional permissions:

All parent container objects for the original and new zones	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read name Read Name
The serviceConnectionPoint object for the computer account	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Write name Write Name **Note** The Name property is the common name (cn) of the serviceConnectionPoint object.
Original parent container for the computer account in the current zone For example, if you are moving a computer from the Finance zone to the Corporate zone, the target object would be: domain/UNIX/Zones/Finance/Computers	On the **Object** tab, select **Allow** to apply the following permission to this object only: Delete serviceConnectionPoint Objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID
New parent container for the computer account in the new zone For example, if you are moving a computer from the Finance zone to the Corporate zone, and you use the default Computers container, the target object would be: domain/UNIX/Zones/Corporate/Computers	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID

Note: You can set the permissions for modifying computer accounts by clicking the **Security** tab when you are viewing a computer's properties.

Preparing a Computer Object

To prepare a computer account in a zone before joining, the following permissions apply to the user or group you want to designate as the trustee for joining the domain.

The serviceConnectionPoint object for the computer account	Click the **Object** tab and select **Allow** to apply the following permission to this object only: Read all properties Write keywords property Write displayName property
Computer account object in Active Directory For example, if the computer account name is AJAX: domain/Computers/AJAX	Click the **Object** tab and select **Allow** to apply the following permission to this object only: Read Permission Reset Password Write userAccountControl Validated write to DNS host name Validated write to service principal name Write to service principal name Write msDS-SupportedEncryptionTypes Write Account Restrictions Write Description Write displayName Write computer name (Pre-Windows 2000) Delete Delete Subtree All Extended Rights

The adjoin command resets the computer account and grants the computer's SELF account the following permissions:

- Write operatingSystem
- Write operatingSystemVersion
- Write operatingSystemHotfix
- Write operatingSystemServicePack
- Write altSecurityIdentities

Creating The Computer Object Manually

If you use Active Directory Users and Computers to prepare the computer object instead of the Prepare Computer wizard, the following permissions must be granted on the computer for the trustee:

The serviceConnectionPoint object for the computer account	Click the **Object** tab and select **Allow** to apply the following permission to this object only: Read all properties Write keywords property Write displayName property
Computer account object in Active Directory For example, if the computer account name is AJAX: domain/Computers/AJAX	Click the **Object** tab and select **Allow** to apply the following permission to this object only: Read Permission Reset Password Write userAccountControl Validated write to DNS Host Name Validated write to service principal name Write Account Restrictions Write Description Write displayName Write computer name (Pre-Windows 2000) Write operatingSystem Write operatingSystemVersion Write operatingSystemHotfix Write operatingSystemServicePack Write altSecurityIdentities Write msDS-SupportedEncryptionTypes Delete Delete Subtree All Extended Rights

Modifying Computer Roles

If you use computer role assignments to control access to a computer, the following permissions are required to modify computer roles:

msDS-AzScope This object is listed under a globally unique identifier (GUID) for the Authorization object. For example: CN=cab186af-61a0-4d54-a0dd...	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read description Read msDS-AzScopeName Read msDS-AzApplicationData Write description Write msDS-AzScopeName Write msDS-AzApplicationData

Deleting Computer Roles

If you use computer role assignments to control access to a computer, the following permissions are required to delete computer roles:

msDS-AzScope This object is listed under a globally unique identifier (GUID) for the Authorization object.	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read Name Read name Read displayName Allow Delete Allow Delete Tree

Setting Permissions For Zone Users

The specific objects and permissions required to work with user accounts depend on the type of zone the user account is associated with and the task to be performed.

In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard instead of assigning the permissions manually.

Adding Users To Standard Zones

In a standard Centrify zone when the functional level of the forest is Windows Server 2003 or later, adding a user account with an Active Directory security group as the primary group to a zone requires the following permissions:

Parent container object for the user profile For example, if you use classic zones, the default Users container in the Finance zone: domain/UNIX/Zones/Finance/Users	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects This permission is required for both standard zones and RFC 2307compliant zones. For standard zones, you need to apply additional permissions. Click the **Properties** tab and select **serviceConnectionPoint objects** from the object list, then select **Allow** to apply the following properties to this object: Read Name Read name Read displayName
User account object in Active Directory For example: domain/Users/user_name	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectCategory Read objectClass Read objectGUID Read objectSid Read userAccountControl
Parent container object for the individual zone For example, if you are adding a user to the Finance zone: domain/UNIX/Zones/Finance	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID Write Description

Modifying Users In Standard Zones

In a standard zone, modifying user account properties for a user with a standard Active Directory security group as the primary group requires the following permissions:

The serviceConnectionPoint object for the user account For example, if you are using classic zones and the UNIX user name is chris: domain/UNIX/Zones/Finance/Users/chris then select serviceConnectionPoint objects	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributesEffective Read objectGUID Write keywords If you are changing the UNIX user name for the user, you need the following additional permissions applied to this object: Read name Write name Write Name property **Note** The Name property is the common name (cn) of the serviceConnectionPoint object.

Note: You can set the permissions for modifying user accounts by clicking **Permissions** when you are viewing the Centrify Profile for a user.

Modifying Users In Rfc 2307-compliant Zones

In a standard RFC 2307-compliant zone, modifying user account properties for a user with an Active Directory security group as the primary group requires the following permissions:

The serviceConnectionPoint object for the user account For example, if you are using classic zones and the UNIX user name is chris: domain/UNIX/Zones/Finance/Users/chris then select serviceConnectionPoint objects	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributesEffective Write keywords Write uid Write uidNumber Write gidNumber Write loginShell Write unixHomeDirectory If you don't see some of these attributes listed for serviceConnectionPoint objects, change the object selected to **posixAccount objects**, then click **Allow** for the additional properties. The GECOS field in a user's UNIX profile is derived from the displayName attribute or the Name property (cn).

> **Note**: You can grant the required permissions to specific users or groups for any zone by selecting the **Modify users** task in the Zone Delegation Wizard.

Listing Users In Standard Zones

In a standard zone, listing user account information requires the following permissions:

The serviceConnectionPoint object for the user account	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each user included in the list: Read displayName Read managedBy Read objectClass Read Name to display the UNIX name Read keywords to display the other UNIX attributes

Listing Users in RFC 2307-Compliant Zones

In a standard RFC 2307-compliant zone, listing user account information requires the following permissions:

The serviceConnectionPoint object for the user account	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each user included in the list: Read displayName Read keywords Read managedBy Read objectClass Read uid to display the UNIX name Read uidNumber to display the UNIX UID Read gidNumber to display the GID of the user's primary group Read logonShell to display the default shell for the user Read unixHomeDirectory to display the user's home directory Read Public Information to display the userPrincipalName for the user

Removing Users from Zones

Removing a user account from a standard zone or RFC 2307-compliant zone requires the following permission:

The serviceConnectionPoint object for the user account	On the **Object** tab, select **Allow** to apply the following permission to this object only: Delete

Setting Permissions for Zone Groups

The specific objects and permissions required to work with group accounts can vary depending on the type of zone the group is associated with and the task to be performed.

In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard instead of assigning the permissions manually.

Adding Security Groups to Zones

Adding an Active Directory group to a zone requires the following permissions:

Parent container object for the group For example, if you are using classic zones, the ZoneName/Groups container: domain/UNIX/Zones/acme/Groups	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass **Note** You can grant the required permissions to specific users or groups by selecting the **Add or remove groups** task in the Zone Delegation Wizard.
Group account object in Active Directory For example: domain/UNIX/UNIX groups/group_name	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read groupType Read objectCategory Read objectClass Read objectGUID Read objectSid
Parent container object for the individual zone For example, if you are adding a group to the Finance zone: domain/UNIX/Zones/Finance	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID Write Description

Modifying Groups in Standard Zones

In a standard zone, modifying a group profile in a zone requires the following permissions:

The serviceConnectionPoint object for the group account For example, if the UNIX group name is web-qa in the HKLab zone: domain/UNIX/Zones/HKLab/Groups/web-qa then select serviceConnectionPoint objects	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributesEffective Read objectGUID Read Name If you are changing the UNIX group name for a group, you need the following additional permissions applied to this object: Read name Write name Write Name **Note** The Name property is the common name (cn) of the serviceConnectionPoint object.

Modifying Groups in RFC 2307-Compliant Zones

In a standard RFC 2307-compliant zone, modifying a UNIX-enabled group in a zone requires the following permissions:

The serviceConnectionPoint object for the group account	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read allowedAttributesEffective Read objectGUID Read Name If you are changing the UNIX group identifier for a group, you need the following additional permissions applied to this object: Read gidNumber Write gidNumber **Note** If you don't see this attribute listed for the serviceConnectionPoint object, change the object selected to **posixGroup objects**. If you are changing the UNIX name for a group, you need the following additional permissions applied to this object: Read name Write name Write Name **Note** The Name property is the common name (cn) of the serviceConnectionPoint object.

Listing Groups in Zones

In a standard zone, listing group account information requires the following permissions:

The serviceConnectionPoint object for the group account	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each group included in the list: Read displayName Read managedBy Read objectClass Read Name to display the UNIX group name Read keywords to display the UNIX GID

Listing Groups in RFC 2307-Compliant Zones

In a standard RFC 2307-compliant zone, listing group account information requires the following permissions:

The serviceConnectionPoint object for the user account	Click the **Properties** tab and select **Allow** to apply the following properties to this object for each user included in the list: Read displayName Read keywords Read managedBy Read objectClass Read objectGUID Read Name to display the group name Read gidNumber to display the group GID

Removing Groups from Zones

Removing an Active Directory group from a standard zone or RFC 2307-compliant zone requires the following permission:

The serviceConnectionPoint object for the group account	On the **Object** tab, select **Allow** to apply the following permission to this object only: Delete

Setting Permissions for License Keys

Starting Access Manager requires the following permissions on the container object for licenses:

The domain root object For example, if the root domain of the forest is arcade.com: DC=arcade,DC=com	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass
Parent container for the Licenses container object For example: domain/Centrify UNIX	On the **Object** tab, select **Allow** to apply the following permission to this object only: List contents
Parent container for license keys For example, the Licenses container object you created or selected in the Setup Wizard: domain/Centrify UNIX/Licenses	On the **Object** tab, select **Allow** to apply the following permission to this object only: List contents

To add and remove license keys, your user account must have the following permissions:

Parent container for license keys For example, the Licenses container object you created or selected in the Setup Wizard: domain/Centrify UNIX/Licenses	Click the **Properties** tab and select **Allow** to apply the following properties to this object and all child objects: Write Description

Setting Permissions for NIS Maps

You can delegate administrative permissions for all NIS maps in a zone or for any specific NIS map within a zone by selecting either the NIS Maps parent container object or the specific NIS map object you want to work with. If you select the NIS Maps parent container object, the permissions you set apply to all NIS maps you add to the zone. If you select the individual NIS map object, the permissions you set only apply to that individual NIS map.

To set permissions on NIS maps or NIS map entries

1. Open the ADSI Edit MMC snap-in and connect to the Active Directory domain.

   Note: For NIS maps, you must use the Zone Delegation Wizard or ADSI Edit to set Active Directory permissions.

2. In the console tree, navigate to the zone folder.

   For example, if you deployed using the recommended organizational structure, expand the domain, Centrify, Zones, and select a specific zone name.

3. Select **CN=NisMaps** to set permissions for all NIS maps in a zone, right-click, then select **Properties**.

   If setting permissions for an individual map, expand CN=NisMaps, then select the map object—such as CN=auto_master—right-click and select **Properties**.

4. Click the Security tab, then click **Advanced**.
5. Click **Add** to search for the user or group to which you want to give administrative privileges, select the user or group in the results, then click **OK**.
6. Scroll to locate the appropriate permissions for the object and its properties to allow the selected user or group to perform the administrative task, click **Allow**, then click **OK**.

In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard instead of assigning the permissions manually.

Adding NIS Maps to a Zone

To add NIS maps to the NIS Maps parent container in a zone, the user account must have the following permissions:

Parent container for NIS Maps For example, if you are using classic zones: domain/UNIX/Zones/ZoneName/NISMaps	On the **Object** tab, select **Allow** to apply the following permissions to this object and all child objects: Create Container Objects

Deleting NIS Maps from a Zone

To delete NIS maps in a zone, the user account must have the following permissions:

Parent container for NIS Maps	On the **Object** tab, select **Allow** to apply the following permissions: Delete Container Objects applied to this object and all child objects. On the **Object** tab, set Apply onto to **Container objects**, then select **Allow** to apply the following permissions: Delete Subtree **Note** This permission is required if the map contains any entries.

Adding Map Entries to NIS Maps

To add entries to any NIS map in a zone, the user account must have the following permissions:

Parent container for NIS Maps	On the **Object** tab, set Apply onto to **Container objects**, then select **Allow** to apply the following permissions: Create classStore Objects

Modifying Map Entries in NIS Maps

To modify entries in any NIS map in a zone, the user account must have the following permissions:

Parent container for NIS Maps	Click the **Properties** tab, set Apply onto to **classStore objects**, then select **Allow** for the following properties: Write adminDescription Write Description Write wWWHomePage

Changing the Map Type for NIS Maps

To change the map type for any NIS map in a zone, the user account must have the following permissions:

Parent container for NIS Maps	Click the **Properties** tab, set Apply onto to **This object and all child objects**, then select **Allow** for the following properties: Write Description

Deleting Map Entries from NIS Maps

To delete entries from any NIS map in a zone, the user account must have the following permissions:

Parent container for NIS Maps	Click the **Properties** tab, set Apply onto to **classStore objects**, then select **Allow** for the following properties: Write name Write Name

Adding Entries to a Specific NIS Map

To add entries to a specific NIS map in a zone, the user account must have the following permissions:

Individual NIS map	On the **Object** tab, select **Allow** to apply the following permissions to this object and all child objects: Create classStore Objects

Modifying Entries in a specific NIS Map

To modify the entries in a specific NIS map in a zone, the user account must have the following permissions:

Individual NIS map	Click the **Properties** tab, set Apply onto to **classStore objects**, then select **Allow** for the following properties: Write adminDescription Write Description Write wWWHomePage

Changing the Map Type for a Specific NIS Map

To change the map type for a specific NIS map in a zone, the user account must have the following permissions:

Individual NIS map	Click the **Properties** tab, set the Apply onto to **This object and all child objects**, then select **Allow** for the following properties: Write Description

Deleting Map Entries from a Specific NIS Map

To delete entries from a specific NIS map in a zone, the user account must have the following permissions:

Individual NIS map	Click the **Properties** tab, set Apply onto to **classStore objects**, then select **Allow** for the following properties: Write name Write Name

Setting Permissions for Password Synchronization

If you want to use the Network Information Service, adnisd, and the Centrify Password Filter to support "agentless" authentication of NIS client requests, the computer that will service the requests must be a member of the zone_nis_servers group in the zone and must be able to access the Active Directory attribute that stores the password hash. The specific permissions required depend on the attribute being used to store the password hash.

Centrify Password Synchronization Service

If you are using the Centrify Password Filter synchronization service, the zone_nis_servers group requires the following permissions:

altSecurityIndentities	Read altSecurityIndentities
msSFU30Password	Read msSFU30Password
unixUserPassword	Read unixUserPassword All Extended Rights

Microsoft Password Synchronization Service

If you are using the Microsoft password synchronization service and the Centrify Network Information Service, adnisd, to authenticate NIS client requests, you must set the following permissions at the domain level, on the Users container object, or on another container that applies to all users.

Users container or a container that applies to all users	Click the **Object** tab, set the Apply onto to **User objects** and select **Allow** to apply the following permission: All Extended Rights You can apply this permission to Domain Computers or to a specific group of computers that contains the computer where the adnisd service is running.

For information about installing and configuring a Microsoft password synchronization service, see the Microsoft documentation for that service or refer to documentation on the Microsoft Web site.

Setting Permissions for Rights and Roles

If you define specific rights and establish role-based access controls on a zone-by-zone or computer-by-computer basis, you might want to manually assign permissions for users who can configure rights and roles.

In most cases, you can grant the required permissions to specific users or groups by selecting the appropriate task in the Zone Delegation Wizard instead of assigning the permissions manually.

Creating the Authorization Store

All of the information about rights, roles, and role assignments is held in an **authorization store** for each zone in Active Directory. The name of authorization store object is CN=Authorization under the zone object's DN. For example, the authorization store for the zone named EMEA_Territories in the Arcade.Net forest is:

cn=Authorization, cn=EMEA_Territories, cn=Zones, cn=UNIX, dc=Arcade, dc=Net

To create the authorization store for a zone, users must have the following permissions:

Parent container for an individual zone For example, a ZoneName container object, such as: domain/Centrify/Zones/arcade	On the **Object** tab, select **Allow** to apply the following permissions to this object and all child objects: List contents Read all properties Read Permissions Select **Allow** to apply the following permissions to this object only: Create msDS-AzAdminManager objects

Defining Rights And Roles in the Authorization Store

To configure rights, roles, and role assignments, users must have the following permissions for the authorization store:

Authorization	On the **Object** tab, select **Allow** to apply the following permissions to this object and all child objects: List contents Read all properties Write all properties
msDS-AzApplication This object is listed under a globally unique identifier (GUID) for the Authorization object. For example: CN=cab186af-61a0-4d54-a0dd...	On the **Object** tab, select **Allow** to apply the following permissions to this object (listed as CN=GUID under the Authorization object) and all child objects: Create and delete msDS-AzOperation objects Create and delete msDS-AzTask objects Create and delete msDS-AzRole objects Create msDS-AzScope objects **Note** You must grant these permissions on the CN=GUID object if you are granting permissions manually with ADSI Edit. The proper permissions are set automatically for the users when you delegate administrative tasks for a zone.

Configuring Authorization in Classic Zones

Unlike hierarchical zones, authorization is an optional feature in classic zones. You must be an administrator or the user who created a classic zone to initialize the authorization store in Active Directory, identify the users who should be allowed to configure rights, roles, and role assignments, and enable or disable the enforcement of the rights and role assignments you have configured.

To update the list of users and groups who are allowed to configure DirectAuthorize rights and roles, you must have the Modify permissions right on the Authorization container under the classic zone container applied to this object and all child objects. If you have this permission, you can click **Add** to add Windows users and groups to the list of users and groups who can configure rights and roles. If you have the Modify permissions right, you can also select a user or group in the list and click **Remove** a user or group from the list.

Adding Roles

To add roles for users or groups, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzTaskObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: Create msDS-AzTask objects Click the **Properties** tab, then select **Allow** for the following properties: Read objectClass

Modifying Roles

To modify roles for users or groups, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzTaskObjectContainer/CN=roleName This object is listed under a globally unique identifier (GUID) for the Authorization object and a specific role name.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Read description Read msDS-AzApplicationData Write Name Write name Write description Write msDS-AzApplicationData

Deleting Roles

To delete roles for users or groups, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzTaskObjectContainer/CN=roleName This object is listed under a globally unique identifier (GUID) for the Authorization object and a specific role name.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Allow Delete

Adding Rights

To add the definition for a right in a zone, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-OpObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: Create msDS-AzOperation objects Click the **Properties** tab, then select **Allow** for the following properties: Read objectClass

Modifying Rights

To modify right definitions in a zone, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzTaskObjectContainer/CN=roleName This object is listed under a globally unique identifier (GUID) for the Authorization object and a specific role name.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Read description Read msDS-AzApplicationData Write Name Write name Write description Write msDS-AzApplicationData

Deleting Rights

To delete right definitions in a zone, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-

	AzApplicationData
msDS-AzOpObjectContainer/CN=pamrightName or msDS-AzOpObjectContainer/CN=pcrightName This object is listed under a globally unique identifier (GUID) for the Authorization object and a specific PAM access right name or privileged command name.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Allow Delete

Adding or Removing Rights from Roles

To add or remove rights from a role in a zone, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzTaskObjectContainer/CN=roleName This object is listed under a globally unique identifier (GUID) for the Authorization object and a specific role name.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Read msDS-OperationsForAzTask Write msDS-OperationsForAzTask

Adding Role Assignments

To add a role assignment, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzRoleObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: Create msDS-AzRole objects

Modifying Role Assignments

To modify role assignments, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzRoleObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: Create msDS-AzRole objects
msDS-AzRoleObjectContainer/CN=CRA_guid This object is listed under a globally unique identifier (GUID) for the Authorization object and a unique identifier for the role assignment.	Click the **Properties** tab, then select **Allow** for the following properties to allow changes to the assigned user or groups: Read Name Read name Allow Delete Click the **Properties** tab, then select **Allow** for the following properties to allow changes to the available time for a role assignment: Read Name Read name Read msDS-AzApplicationData Write msDS-AzApplicationData

Deleting Role Assignments

To modify role assignments, users must have the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-AzRoleObjectContainer/CN=CRA_guid This object is listed under a globally unique identifier (GUID) for the Authorization object and a unique identifier for the role assignment.	Click the **Properties** tab, then select **Allow** for the following properties: Read Name Read name Allow Delete

Setting Permissions for Zone Provisioning

The Zone Provisioning Agent requires permission to create UNIX profiles, that is, the service connection points in each zone where it needs to perform provisioning operations. The service account that runs the Zone Provisioning Agent requires the Log on as a service right set as a local computer security policy, or in the default domain policy.

Supplemental Installation Notes

This document includes various notes about installing Server Suite on different operating system platforms.

Verifying the DNS Configuration on Linux

The Server Suite Authentication Service (DirectControl) uses DNS to locate domain controllers for the Active Directory forest. To verify the Active Directory domain controller can be located through DNS, try sending a ping request to the computer.

You can also run the adinfo --diag command to attempt to read the DNS records for the domain you want to join. For example:

adinfo --diag domain_name

If DNS is properly configured, the command should display the LDAP URLs for the domain controllers in the domain you want to join.

For more detailed information about configuring DNS or troubleshooting your DNS configuration, see the *Administrator's Guide for Linux and UNIX*.

Joining the Domain (Zoned Mode Only)

To join an Active Directory domain manually:

1. On the Linux computer, log in as or switch to the root user.
2. Run adjoin to join an existing Active Directory domain using a fully-qualified domain name.

   adjoin --zone <zone_name> --user <user_name><domain_name>

   The user account you specify must have permission to add computers to the specified domain and zone. If you don't specify a user name, the Administrator account is used by default.
3. Type the password for the specified user account.

   If the authentication service can connect to Active Directory and join the domain, a confirmation message is displayed. You can now enable existing Active Directory groups and users to work with this Unix computer.

For more information about the options you can specify when joining a domain, see the man page for the adjoin command or the *Administrator's Guide for Linux and UNIX*.

To step through common tasks and test scenarios, see the *Evaluation Guide for Linux and UNIX*.

Joining the Domain (Express mode)

To join an Active Directory domain manually:

1. On the UNIX computer, log in as or switch to the root user.
2. Run adjoin to join an existing Active Directory domain using a fully-qualified domain name.

   adjoin --workstation --user <user_name> <domain_name>

   The user account you specify must have permission to add computers to the specified domain. If you don't specify a user name, the Administrator account is used by default.
3. Type the password for the specified user account.

   If the authentication service can connect to Active Directory and join the domain, a confirmation message is displayed.

   For more information about the options you can specify when joining a domain, see the man page for the adjoin command or the *Administrator's Guide for Linux and UNIX*.

HPUX Installation Notes

This section describes the unique characteristics or known limitations that are specific to using authentication service on a computer with the HP-UX operating environment.

ia64 - Mapping Local HP-UX User Accounts to Active Directory Accounts

In most environments, you can map local user accounts to Active Directory accounts to manage the passwords for local users using your Active Directory password policies. On HP-UX, however, if an account is a valid Active Directory account but the authentication through Active Directory fails, the PAM module will attempt to authenticate the account locally and will allow the account to log on if the local authentication succeeds. Because users can still log on to HP-UX systems using their local account password, you cannot effectively use Active Directory or the User Map group policy to enforce your password policies for local HP-UX user accounts.

To enforce Active Directory password policies for local HP-UX users, you need to disable the local user accounts to prevent those local account names and passwords from being used to log on.

Entering an Incorrect Password on HP-UX

On HP-UX, if Server Suite-enabled users enter an incorrect password, they are normally prompted with a second "System password" prompt. This prompt is asking for a password for a local user, regardless of whether that user actually exists locally on the system. If the user exists locally, this prompt allows the user to log in using the local password. If the user does not exist locally, this prompt is unnecessary and will not allow the authentication service-enabled user to log in, regardless of the password entered.

This second prompt can be avoided by changing the options in /etc/pam.conf to the authentication modules. Two changes are necessary:

1. Add an option to the authentication service PAM module to prompt all users for a password (not just Active Directory users)
2. Add an option to the HP-UX UNIX login module to use the password obtained by the authentication service module.

   The lines which need to be modified appear like this in the file:

   service_name auth sufficient /usr/lib/security/libpam_centrifydc.1 debug

   service_name auth required /usr/lib/security/libpam_unix.1

   Where service_name is something like login, dtlogin, ftp, or similar. The pam_centrifydc.1 line needs the ask flag to prompt all users for passwords. The libpam_unix.1 line needs the use_first_pass option. For example:

   login auth sufficient /usr/lib/security/libpam_centrifydc.1 debug ask

   login auth required /usr/lib/security/libpam_unix.1 use_first_pass

   > **Note**: It is extremely important that the pam_centrifydc line appear before the pam_unix line in the file, or users will never be prompted for a password. Administrators should be extremely careful when editing this file. Any typographically errors in this file could prevent all users from logging on to the system and render the system unusable.

AIX Installation Notes

Support for AIX Capabilities Attribute

Support has been added for the AIX Capabilities user attribute, a feature that is only available on AIX 5.3 and later. To enable the feature, edit /etc/centrifydc/centrifydc.conf to add the following line:

lam.method.version: 520

This allows using methods that are only available with AIX 5.3 and later, and these methods are required to support the Capabilities attribute.

Use adquery to view capabilities for an Active Directory user:

adquery user -X aix.capabilities <ADuser>

Use adupdate to set capabilities for an Active Directory user:

adupdate modify user -X +aix.capabilities=CAPABILITIES <ADuser>

Where CAPABILITIES is a comma-separated list of capabilities to add for the user. For example:

CAP_NUMA_ATTACH, CAP_BYPASS_RAC_VMM, CAP_PROPOGATE

Currently there is no group policy support for capabilities, this may be implemented in a future release of authentication service.

Users Cannot Log in by way of FTP if They Have a Restricted Shell

On AIX 6.1, a user's login shell must appear in the shells attribute of the /etc/security/login.cfg file. Delinea Privilege Elevation Service does not add dzsh to this attribute so by default an ftp user who is using dzsh as their login shell cannot log in. To workaround this issue, add /usr/bin/dzsh to the shells attribute of /etc/security/login.cfg.

Starting and Stopping DirectControl on AIX

Because the authentication service daemon, adclient, is defined as an AIX system resource, you use the following commands to start, stop, and check the status of the daemon:

startsrc -s centrifydc

stopsrc -s centrifydc

lssrc -s centrifydc

Using the Server Suite Authentication Service LDAP Proxy on AIX

When using the LDAP Proxy on AIX you need the following line in the slapd configuration file at

/usr/share/centrifydc/etc/openldap/ldapproxy.slapd.conf
moduleload /usr/share/centrifydc/libexec/openldap/libback_centrifydc.a(libback_centrifydc.so.0)

> **Note**: This should be entered as a single line into the configuration file. This line may already be in the configuration file, but commented out, in which case you can just remove the leading "#" to uncomment it.

Setting the DNS Configuration Parameter to Join the Domain on SuSE Linux

To successfully join a Active Directory domain on computers running SuSE Linux, you must set the mdns option to off in the /etc/host.conf file. If your /etc/host.conf file does not include the following line, you should add it to the file:

mdns off

This setting is required to enable proper DNS resolution, and therefore, must be set to successfully join the domain, and to allow users to log on properly.

Mounting CIFS Shares

Common Internet File Systems (CIFS) provides an open and cross-platform protocol for requesting remote network server files and services. When a CIFS share is mounted on a Centrify Linux system, file ownership is listed incorrectly.

To correct this, apply the CentrifyDC-cifsidmap plug-in. The CentrifyDC-cifsidmap plug-in enables mapping AD User/Group Security IDs (SIDs) to User/Group IDs (UIDs/GIDs) configured in a zone and from UIDs/GIDs to AD User/Group SIDs correctly. This, in turn, allows the CIFS Client integration with DirectControl.

Use Cases

Mapping UIDs to SIDs is not always required when mounting CIFS shares. But it is needed when working with the files on the shares. For example, when modifying Access Control Links (ACLs). In version 5.8 and older, the cifs-utils package uses the winbind daemon for this mapping. Through winbind, the /usr/sbin/cifs.idmap binary was linked against libraries.

The /usr/sbin/cifs.idmap binary works in conjunction with the Samba winbind facility to map owner and group SIDs to UIDs and GIDs respectively.

With version 5.9 the winbind facility does not perform this mapping. Use the **CentrifyDC-cifsidmap** plug-in to ensure that:

- cifs-idmap translates the ownership on the SMB share correctly.
- the kernel determines who has rights to the CIFS share mount directories and files correctly.
- AD User/Group SIDs are mapped correctly and all the IDs are consistent and correct.

For example:

To see the incorrect file ownership: mount your CIFS share and display the ownership of the files in the mounted share.

1. Mount the share. This command requires root privileges.

   Syntax:

   sudo mount -t cifs *domain_ip/path /local/path/* -o username=*your_user_name*, *file_privilege*, password=*your_password*, domain=*domain_name*, cifsacl

   Example:

   sudo mount -t cifs //192.168.0.100/cifsshare /tmp/mntshare1/ -o username=cifsdemouser1,rw, password=My1Pass,domain=example.com,cifsacl

   The cifs type (-t cifs) requires the cifsacl option. See man mount.cifs for command usage.

2. List all the files on the mounted file system.

   If the CIFS share is owned by root, then you need to use sudo to view the files on the mounted directory, because the files you are verifying can only be seen with root privileges.

   sudo ls -al /mntshare1

   ... ... root root ... cifsdemouser1.txt
   ... root root ... cifsdemouser2.txt
   ... root root ... cifsdemouser3.txt

   If the AD user names are not listed, and only root is listed at the owner of the files, then you need to install the CentrifyDC-cifsidmap plug-in. Complete the steps in the following sections.

CentrifyDC-cifsidmap Plug-in Requirements

The Centrify CIFS idmap plug-in is available only for supported systems. The cifs.idmap-plugin requires:

Operating system versions:

- RedHat 7 or above
- Debian 8 or above

- SUSE 12 or above

cifs-utils version:

- cifs-util 5.9 or above

Prepare to Install the CentrifyDC-cifsidmap Plug-in

Prior to installing the CentrifyDC-cifsidmap plug-in, install and configure the following:

- Install the cifs-utils

  The cifs-utils are a package of tools used on CIFS filesystems. See your CIFS documentation.

- Install CentrifyDC

  See the Planning and Deployment Guide.

- Join the machine to AD

  See the Planning and Deployment Guide.

Install the CentrifyDC-cifsidmap Package

1. Verify cifs-utils package is installed. Install it, if it is not already installed. For example:

   It is possible to manually configure your system without cifs-utils, but the program /usr/sbin/cifs.idmap, is still required for the Centrify CIFS idmap plug-in to work.

   - SUSE or RedHat

     yum install cifs-utils

   - Debian

     apt-get install cifs-utils

2. Download the CentrifyDC-cifsidmap package and change to the download directory.

   The package contains the cifs-idmap-plugin.

   Example download package names

   - SUSE or RedHat

     CentrifyDC-cifsidmap-5.5.1-rhel5.x86_64.rpm

   - Debian

     CentrifyDC-cifsidmap-5.5.1-deb8-x86_64.deb

     Example download directory

     # cd /home/user1/Download/

3. Run the appropriate package install command from the download directory.

   - SUSE or RedHat

     # sudo rpm -i CentrifyDC-cifsidmap-5.5.1-rhel5.x86_64.rpm

   - Debian

     # sudo dpkg -i CentrifyDC-cifsidmap-5.5.1-deb8-x86_64.deb

4. Verify the CentrifyDC-cifsidmap package is installed correctly. Check the libcifsidmap.so location.

- SUSE or RedHat

  # ls /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so -al

  ... /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so

- Debian

  # ls /usr/share/centrifydc/lib/plugins/cifs/libcifsidmap.so -al

  ... /usr/share/centrifydc/lib/plugins/cifs/libcifsidmap.so

Configure cifs-utils for CentrifyDC-cifsidmap Plug-in

On Linux, the command, alternatives, is a tool for managing different software packages that provide the same functionality. The alternatives command, on different systems has different names and locations. For additional information on alternatives use, see your Linux documentation.

- RedHat

  /usr/sbin/alternatives

- SUSE and Debian

  /usr/sbin/update-alternatives

To configure the cifs-utils

1. Check the status of /etc/cifs-utils/idmap-plugin and note the priority level.

   For example on RedHat:

   # pwd

   /etc/cifs-utils

   # ls -al

   ... idmap-plugin -- /etc/alternatives/cifs-idmap-plugin

   # alternatives --display cifs-idmap-plugin

   ...
   /usr/lib64/cifs-utils/cifs_idmap_sss.so - priority 20
   ... Current 'best' version is /usr/lib64/cifs-utils/cifs_idmap_sss.so.

   In this example the cifs_idmap_sss.so plugin object has the highest priority and that priority is set to 20.

2. Configure cifs-utils to use the CentrifyDC-cifsidmap plug-in, cifs-idmap-plugin.

   Run the commands appropriate for your OS.

   Include a priority that is higher than the priority listed in Step 1. For example, the priority in Step 1 is 20, set this cifs-idmap-plugin priority to 21 or higher.

   - RedHat

     alternatives --install /etc/cifs-utils/idmap-plugin cifs-idmap-plugin /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so <*priority*>

     alternatives --set cifs-idmap-plugin /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so

   - SUSE or Debian

     update-alternatives --install /etc/cifs-utils/idmap-plugin cifs-idmap-plugin /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so <*priority*>

     update-alternatives --set cifs-idmap-plugin /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so

3. Verify the CentrifyDC-cifsidmap plug-in is configured correctly. Run the appropriate alternatives display option.

- RedHat

  alternatives --display cifs-idmap-plugin

- SUSE or Debian

  update-alternatives --display cifs-idmap-plugin

4. Verify the cifs-idmap-plugin location and priority. Review the alternative command response.

   The cifs-idmap-plugin priority needs to be higher than other listed idmaps. The Current 'best' version needs to point to the cifs-idmap-plugin location.

   For example on RedHat:

   # alternatives --display cifs-idmap-plugin

   ... /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so - priority 21
   Current 'best' version is /usr/share/centrifydc/lib64/plugins/cifs/libcifsidmap.so.

Mount the CIFS Share and Confirm File Ownership

Only mount CIFS shares as root user or use sudo.

1. Verify the receiving mount directory. Create a directory to receive the mount files, if you do not have one. For example:

   cd /tmp
   mkdir mntshare1
   ls -al /mntshare1

2. Optionally, verify that the user(s), you are expecting to be owners of CIFS shared files, are valid AD users. For example:

   adquery user cifsdemouser1

   cifsdemouser1:x:1019226236:1019226232:cifsdemouser1:home/cifsdemouser1:/bin/bash

3. If you previously mounted the CIFS share, and found that file ownership was incorrect, unmount it now. For example:

   sudo unmount /tmp/mntshare1/

4. Mount the share. This command requires root privileges.

   Syntax:

   sudo mount -t cifs *domain_ip/path /local/path/* -o username=*your_user_name*, *file_privilege*,
   password=*your_password*, domain=*domain_name*, cifsacl

   Example:

   sudo mount -t cifs //192.168.0.100/cifsshare /tmp/mntshare1/ -o username=cifsdemouser1,rw,
   password=My1Pass,domain=example.com,cifsacl

   The cifs type (-t cifs) requires the cifsacl option. See man mount.cifs for command usage.

5. List all the files on the mounted file system.

   If the CIFS share is owned by root, then you need to use sudo to view the files on the mounted directory, because the files you are verifying can only be seen with root privileges.

   sudo ls -al /tmp/mntshare1

   ... ... cifsdemouser1 root ... cifsdemouser1.txt
   ... cifsdemouser2 root ... cifsdemouser2.txt
   ... cifsdemouser3 root ... cifsdemouser3.txt

   Notice the AD users are listed as owners of the CIFS share files. This completes the task.

Delinea

Known Issues

Here are some known issues, organized by category.

Installation and Un-installation Issues

- Upgrading from the beta build to this version may result in offline MFA mode if there are multiple authentication servers registered in your AD forest. To resolve this, uninstall the beta build first and then install this new version. (Ref: CS-41915)
- The Centrify Common Component should be the last Server Suite component uninstalled. If the component is uninstalled before other component, it must be reinstalled by the uninstall process to complete its task. (Ref: 36226a)
- If you intend to install the software on the desktop with elevated privilege, you should not check the "Run with UAC restrictions" option when creating the desktop. (Ref: 39725b)
- When you double-click on the Server Suite Agent for Windows msi and select the "repair" option, the existing files are replaced irrespective of their version number, even when they are identical. As a result, a prompt to restart the system is displayed as files that were in use were replaced. However, if you use the Easy Installer to do the repair and a file on the disk has the same version as the file that is part of the installer package, the installed file will not be replaced. Therefore, there will not be any prompt to restart the system. (Ref: 26561a)
- If you uninstall the Server Suite Agent for Windows while the DirectAudit Agent Control Panel is open, files needed by the uninstall process may be blocked. You should close the DirectAudit Agent Control Panel for a successful conclusion to the uninstall process. (Ref: 25753a)
- Server Suite Agent for Windows and its installer are built on .NET. Therefore, .NET is always installed as a pre-requisite before the agent is installed. If .NET is removed from the system later, Server Suite Agent for Windows will not run properly. User will also experience problem when trying to remove Server Suite Agent for Windows from the system. To properly uninstall Server Suite Agent for Windows, please make sure Server Suite Agent for Windows is uninstalled before .NET. (Ref: 39051a)

Configuration Issues

- In a cross-forest environment, forest A user cannot enroll a device joined to forest B when forest A does not have a connector. (Ref: CS-44805)
- In Windows 2016 and Windows 10, during the login process, selecting SMS or using other mechanisms like Security Question/Phone call/Password/Email/Mobile for MFA and clicking the "Commit" button will be intermittently unresponsive. (Ref: CS-41699)
- In some large environment with multiple domain controllers, it may take up to one minute for the new zone setting in Server Suite Agent Configuration to take effect. (Ref: 58128b)
- If one of the Global Catalog servers is unavailable, user may not be able to configure the zone for Server Suite Agent for Windows. (Ref: 58621b)
- Microsoft normally automatically distributes and installs root certificates to the Windows system from trusted Certificate Authorities (CA) and users are seamlessly able to use a secure connection by trusting a certificate chain issued from the trusted CA. However, this mechanism may fail if the system is in a disconnected environment where access to Windows Update is blocked or this feature of automatic root certificate installation is disabled. Without updates on the certificate trust list (CTL), the default CTLs on the system may not be adequate for secure connections of multi-factor authentication especially for older versions of Windows such as Windows 7 and Windows Server 2008 R2. To ensure the success of multi-factor authentication, user may need manually distribute and install the latest CTLs and the required root certificate to systems in a disconnected environment. See Centrify KB-6724 for further information. (Ref: CS-39703)

Environment Issues

- On Windows 10 and Windows 2016 machines with Centrify Privilege Elevation Service, the following will occur (Ref: CS-43883):
  - Pop up an error dialog several seconds after clicking "Open file location" in the context menu of a shortcut on the start menu. Explorer windows will display correctly.
  - No responses to the following actions
    - Clicking "Open file location" in the context menu of a shortcut on desktop
    - Clicking "Open file location" in the context menu of a shortcut on the Centrify Start menu in the Privileged Desktop
    - Slow response to "OK", "Cancel" in the shortcut property page after "Open file location" in the general tab is clicked. The dialog will close after several seconds.
- On some Windows 10 computers, the smart card login option may not be displayed if another credential method has been recently used. To display the smart card login option, remove and insert a smart card into the reader. This will cause the login screen to reload and will display the smart card login option. (Ref: CS-41282)
- An environment with no Global Catalog is not supported. (Ref: 46577a)
- Centrify Privilege Elevation Service requires machine time to be synchronized with domain controller. VMware virtual machine has a known issue that its time may not be synchronized with domain controller. This problem occurs more often on an overloaded virtual machine host. If the system clocks on the local Windows computer and the domain controller are not synchronized, Centrify Privilege Elevation Service does not allow any domain users to login. You can try the following KB from VMware to fix the time synchronization issue. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1189 (Ref: 47795b)

RunAsRole Issues

- If you use the "RunAsRole.exe /wait" command to run a Python script, the input/output cannot be redirected for versions of Python below 3.0.0. (Ref: 45061a)
- The Run As Role menu is not available on the start screen in Windows 8 or Windows 2012 or later because Microsoft doesn't support any custom context menu on the start screen. User has to go to the Windows desktop in order to launch an application using Run As Role context menu. (Ref: 35487a)
- When running "RunAsRole.exe /wait sc.exe" with no argument provided to sc.exe, sc.exe will prompt
  - Would you like to see help for the QUERY and QUERYEX commands? [y I n]:
  - Typing 'y' or 'n' doesn't do anything because the input cannot be successfully redirected to sc.exe. (Ref: 47016b)
- It is not recommended to change zone via Run As Role since the role that is in use may no longer be available once after leaving from the previous zone during the change zone process. (Ref: 58043a)
- On Windows Server 2008 R2 and Windows 7, if the Agent machine has no internet connection and the .NET CLR settings (checkCertificateRevocationList) is set to True, the MFA authentication will be failed because the CLR is unable to verify the certificate through internet. The workaround is to enable the internet connection or turn off the CLR settings (set checkCertificateRevocationList to False which is also the default value). (Ref: CS-40147)

Desktop with Elevated Privileges issues

- On a desktop with elevated privileges, if you use "Control Panel > Programs > Programs and Features" to uninstall a program, you may see the following warning message and cannot uninstall the software.

  "The system administrator has set policies to prevent this installation."

  This issue happens when User Account Control (UAC) is enabled and when "Run with UAC restrictions" is selected when creating the new desktop. (Ref: 33384a)

- You cannot use the Start menu option "Switch User" while you are using a role-based, privileged desktop. To use the "Switch User" shortcut, change from the privileged desktop to your default Windows desktop. From the default desktop, you can then select Start > Switch User to log on as a different user. (Ref: 39011b)

Roles and Rights Issues

- There is no 'Require multi-factor authentication' system right for the predefined 'Windows Login' role. To define this system right for MFA, use the pre-defined Require MFA for logon role, or create a new custom role. (Ref: CS-40888)
- Windows Network Access rights do not take effect on a Linux or UNIX machines. If you select a role to start a program or create a desktop that contains a Network Access right, you can only use that role to access Windows computers. The Windows computers you access over the network must be joined to a zone that honors the selected role. The selected role cannot be used to access any Linux or UNIX server computers on the network. (Ref: 32980a)
- Network Access rights are not supported on the Windows 2008 R2 Terminal Server if "RDC Client Single Sign-On for Remote Desktop Services" is enabled on the client side. (Ref: 34368b)
- To elevate privileges to the "Run as" account specified in a Windows right, the "run as" account must have local logon rights. If you have explicitly disallowed this right, you may receive an error such as "the user has not been granted the requested logon type at this computer" when attempting to use the right. (Ref: 34266a)
- If your computer network is spread out geographically, there may be failures in NETBIOS name translation. If a NETBIOS name is used, Active Directory attempts to resolve the NETBIOS name based on the domain controller that the user belongs to, which in a multi-segment network might fail. Therefore, Network Access rights might not work as expected if the remote server is located using NETBIOS name. You may need to consult your network administrator to work around this issue. (Ref: 39087a)
- File hash matching criteria in the Application right is not supported for a file larger than 500MB. This is to make sure DirectAuthorize does not spend too much CPU and memory resources to calculate the file hash. User trying to import a file with the size larger than 500MB will see an empty value for the file hash field. (Ref: 56778a)
- For a small set of application, enabled matching criterion - "Product Name", "Product version", "Company", "File Version" or "File Description" of a Windows Application Right may fail to match after upgrading agent under the following conditions: - Any value for the enabled matching criteria is defined by either import from a process or file - The matching criteria is defined by 5.1.3 or 5.2.0 DirectManage Access Manager since the number of affected application is expected to be relatively low, proactively updating the defined matching criteria of Windows Application Right is not necessary. (Ref: 60053a)

Compatibility with Third Party Products Issues

- VirtualDesktop is not compatible with Server Suite Agent for Windows. Users should use the Centrify system tray applet to create virtual desktop instead. (Ref: 44641b)

- The startup path for “SharePoint 2010 Management Shell” and “Exchange Management Shell” may set to C:\Windows instead of user home directory if it is launched via RunAsRole.exe or from a desktop with elevated privilege. (Ref: 38814b, 46943b)
- Attempting to enable Kerberos authentication for Oracle databases will fail. This issue is being brought to the attention of Oracle Support for a resolution in upcoming releases. (Ref: 33835b)
- Some applications do not use the process token to check the group membership. They check the user’s group membership on its own. Therefore, any Windows rights configured to use a privileged group will not take effect in these applications. The workaround is to use a privileged user account instead of a privileged group. Here is the list of known application with this issue:
    - vCenter Server 5.1
    - SQL Server
    - Exchange 2010 or above
    - SCOM 2007

      (Ref: 45318a, 45218a, 43779a, 38016a)
- Users may notice an error and cannot install ActivClient after installing Server Suite Agent for Windows. During the installation of ActivClient, it attempts to change the local security setting. However, there is a known issue for Server Suite Agent for Windows of blocking the local security setting (Ref: 63609b). Therefore, users may not be able to install ActivClient successfully after installing Server Suite Agent for Windows. We suggest installing ActivClient before installing Server Suite Agent for Windows. If Server Suite Agent for Windows has been installed, please uninstall it and follow the installation sequence suggested. This issue happens on Windows 8.1 and Windows 2012 R2 only. (Ref: 76016b)

Application Manager Issues

Application Manager does not support the Server Core edition of Windows. (Ref: CS-40656)

Best Practices

This section, created by Delinea Systems Engineering in collaboration with Delinea Engineering and Delinea Professional Services, describes the deployment best practices for Server Suite. The goal of this document is to outline and document the actions customers can take to prevent unexpected service degradation with the Server Suite product.

Using our best practices that have evolved over many years, we have developed the Server Suite software to be extremely resilient to many types of Active Directory topologies, networks and environments. In addition, we have gathered data from our major deployments to provide the top items that a customer should do to ensure the Server Suite deployment is healthy. The best practices are organized by functional area.

Best Practices For Unix And Linux Systems With Server Suite

This section includes the following topics:

Upgrade Server Suite Agents And Administrative Tools

Many technologies are prone to introducing problems when upgrading to the latest and greatest version. Delinea technology has been around for 15 years and unlike common practice with other technologies of waiting to upgrade, Delinea recommends having the latest and greatest versions installed because these will provide greater stability and security. Delinea provides major releases and minor releases every year. The agent is continually receiving security, performance, and feature updates. The administrator tools (consoles, SDKs, APIs) are continually adding functionality that can be pushed to the agent and more support for automation.

Customers should review security updates from Delinea on a periodic basis.

One of the most important things a customer can do is to upgrade the Server Suite Agent once per year to take advantage of the additional functionality/stability offered with latest versions. Server Suite Agents and administrative tools are easy to upgrade, can be done in a modular fashion and are backwards and forwards compatible. The most recent releases can be found at the Downloads section of the Support Portal.

Customers should leverage Enterprise grade deployment framework (supported by technologies like Chef, Puppet, Bladelogic, etc) to automatically deploy the Server Suite Agent and updates. Leverage Chef/Puppet/BladeLogic to automatically deploy and maintain agent configuration parameters and to leverage the Delinea Repo to automatically upgrade target systems in a streamlined fashion. Another option is the Delinea Software Repo for streamlined installation and updates.

Enable NSCD

Nscd is a daemon that provides a cache for the most common name service requests. The default configuration file, /etc/nscd.conf, determines the behavior of the cache daemon. More information on NSCD can be found here.

We recommend enabling nscd on each Server Suite enabled server to maximize the caching performance. The default configuration of nscd will suffice.

Set Group Policies To Govern The Agent Behavior

One of the most powerful features in the Server Suite platform is the ability to centrally push out Group Policy to Linux systems. We recommend deploying at least 1 GPO to your systems so you have the means to centrally configure the agent behavior in your environment. Group policy settings are documented in the *Group Policy Guide*. In the event a change to the Centrify parameters is needed for the environment, a GPO change can quickly deploy the change to the systems.

Customers should use the “Set crontab entries” GPO to deploy a crontab entry that runs the program /usr/share/centrifydc/adedit/adsyncignore every week. This will assure Server Suite is not looking up in AD local users, not defined in AD. This is a very strong recommendation. Customers can choose another mechanism to deploy the crontab entry to systems if that makes more sense for their environment.

Set agent parameters

Exclusions of Domains

Server Suite provides robust support for complex active directory environments with varying trust relationships. Many agent parameters can be configured through Group Policy. We often see customers don't cleanup decommissioned domains or have domains in the environment not in scope for Server Suite. We recommend blacklisting the domains that are not in scope or whitelisting only the domains in scope for Server Suite.

An example of excluding, black listing, a domain in /etc/centrifydc/centrifydc.conf is:

adclient.excluded.domains: anvil.acme.com

An example of including, white listing, a domain in /etc/centrifydc.conf is:

adclient.included.domains: anvil.acme.com

Paged Control

To operate the best with the Microsoft Active Directory search optimizer, Server Suite provides a parameter called "adclient.schema.extensions.search.add.paged.control". We recommend setting this parameter to true to optimize AD lookups.

Suite 2016.1

If the version of the Server Suite Agent for *NIX running is version 5.3.1, part Suite 2016.1, we highly recommend configuring the parameter "adclient.altupn.update.interval: 90000000" in /etc/centrifydc/centrifydc.conf.

These parameters can be deployed via the GPO"Add centrifydc.conf properties" under Computer Configuration > Centrify Settings > DirectControl Settings. See the *Group Policy Guide* for additional information.

Use the Server Suite DB2 Plugin

DB2 systems normally authenticate users against the local Operating System. Therefore, most customers don't think about performance of authentication and lookups when they centralize authentication to Active Directory. However, as customers centralize authentication to Active Directory, performance considerations become more important since DB2 is very user lookup intensive. Customers that leverage DB2 in their environment should consider using the Server Suite DB2 user and group plugin since it delivers enhanced caching to improves the performance of lookups in a DB2 environment that leverages Active Directory for authentication/authorization.

Best Practices for Active Directory Environment

Index the UID Attribute

Many UNIX applications make requests for the uid attribute as part of their inner workings. If the uid attribute is not indexed and applications make frequent requests for uid data, this can have a negative effect on the performance of Domain Controllers. Centrify highly recommends customers index the uid attribute in Active Directory.

Windows Active Directory functional level and Windows Server version

Customers should maintain an upgrade strategy to use a stable and supported Active Directory functional level and the version of Windows server. As of this writing customers should be moving to Windows 2016 functional modes.

Maintain sites and services domain controller topology

A common issue customers come across is Centrify binding to the wrong Domain Controllers. For example, all the users in the US may be authenticating to a domain controller that is not geographically desirable. In most instances, this occurs because the AD Sites and Services definition does not include the subnets of the UNIX/Linux systems or is not updated on a consistent basis.

A process should be defined where the UNIX/Linux networking teams regularly interact with the AD team to assure subnets are added and removed from AD Sites and Services accordingly.

Centrify Access Model Best Practices

Proper definition of global/child zone structure.

A proper Centrify deployment should have a Global Zone with an appropriate number of Child Zones and Computer Roles to drive access across groups of systems. The general recommendation for defining profiles, roles and rights is:

- UNIX enable all users at the Global Zone level
- In addition, UNIX enable users at the child zone level, if attributes need to be different
- for users on the systems in the child zones (ie.different primary group)
- UNIX enable groups at the Child Zone vs. the Global Zone unless the groups need to be visible across all servers

- Always enable ZPA to automate UNIX profile provisioning across all Zones that will have user/group UNIX profiles
- Define Roles and Rights in the Global zone and assign roles at computer roles or zones if appropriate

A common mistake made is the use of too many Child Zones or use Child Zones incorrectly. Limit child zone sprawl. Child zones should be used for specific purposes like:

Segregating systems in different business units

Segregate the management of groups of systems to different administrative groups

Override the UNIX profiles of users and groups across groups of systems.

Another common mistake is managing roles and rights definition throughout the zone hierarchy which makes it difficult to find roles and rights when updates are needed.

Another mistake is using Zones to define access. Instead, use Computer roles to prevent lateral movement, drive an automated access model and to take advantage of performance benefits. Leverage Computer Roles and AD groups to manage system types by likeness of access and create AD user groups in a similar manner. This promotes automation because user access can be granted access/privileges by simply adding users to the right AD groups. Similarly, systems can be provisioned to the right AD group of computers. Computer roles can be defined by application types. For example, "App 1 DEV" App1 PROD", etc. The goal is to not have to use the access manager UI for provisioning access.

Analyze The Deployment Periodically

As a Server Suite deployment matures, customers should perform a periodic analysis using the Access Manager "Analyze" feature. The analysis highlights problem areas in the Server Suite deployment. For example, the analysis will identify orphaned objects.

Additionally, Centrify recommends periodically reviewing security updates available at our support portal.

Use the Centrify Zone Provisioning Agent

We highly recommend leveraging the Zone Provisioning Agent to automatically provision UNIX profiles for users. Additionally, we recommend two instances of ZPA in large environments. This provides redundancy in the provisioning process. The ZPA service should also be monitored to ensure it is operational.

Deploy Reporting Services and Security Information and Event Management (SIEM)

To maximize the investment, we highly recommend deploying Delinea Reporting Services and SIEM integration. These capabilities provide customers which insight into which users can access which system and security related events the Server Suite Agent reports on. See the following items for more information:

- *Report Administrator's Guide*
- SIEM documentation on the documentation portal
- A Delinea community article

Best Practices for the Audit and Monitoring Service

This section includes the following topics:

Manage the Audit Store Database Size

The Audit Store database needs to be managed according to the company's retention policy which often dictated by the security/compliance team. To assure the audit service performs and scales as required, we recommend keeping the active Audit Store database at most, between 250GB and 500GB in size. Perform a database rotation if your active Audit Store database is larger than 500GB. A database rotation takes the current active database and marks it inactive and makes a new database the active database. See the documentation for how to automate database rotation.

Another approach is to delete audit records and shrink the size of the active database. This approach works well as long as the indexes are also rebuilt. Otherwise, shrinking the database without indexing will lead to fragmented indexes and poor query performance. KB-8472 details how to shrink and re-build the database indexes.

Centrify recommends keeping only databases that are required for auditing purposes attached to the audit infrastructure. The databases that are not needed should be detached. Customers often forget to detach the databases that are outside the company's normal live data/retention policies. Too many attached Audit Store databases result into poor query performance and increased load on the Management database. Periodically review the list of attached Audit

Store databases and detach the ones that are no longer needed to be online as per the retention policy.

Maintain the audit store database index

It is recommended to maintain the audit store database's indexes regularly. This can be done by setting up a simple SQL job to reorganize the indexes if they are 5%-30% fragmented and rebuild the indexes if they're more than 30% fragmented. KB-8472 details how shrink and re-build the database indexes.

Configure SQL Server

There are several SQL specific configurations and server settings that can affect performance and operation.

Avoid deploying the Audit Store databases in a SQL availability group unless it's required by the company's compliance policies.

Configure SQL Server power settings to be set to Balanced instead of High Performance.

SQL Server has a setting called Max Server Memory that controls the maximum amount of physical memor that can be consumed by the SQL Server's buffer pool. An incorrectly configured Max Server Memory may either result into the SQL engine causing high IO or OS/other programs starving for more memory. Refer to the "configuring the maximum memory for audit store databases" section of the Auditing Administrator's Guide and always configure this value as recommended before the deployment begins.

If you're expecting a database server to get migrated/retired in the near future, it's better to create a CNAME alias in DNS for the current database server and specify the alias everywhere (e.g. when creating a new Management database) rather than specifying the actual host name. This will prevent scenarios where the database server is not found after a migration.

Audit and Monitoring Architecture

The audit architecture includes several components to ensure a smooth operating and secure audit environment. A Collector is the service that collects audit records from servers being audited and stores them in the audit store database. Avoid deploying the collector on the same machine as the active Audit Store database's SQL Server.

When using the Server Suite Agent for Windows to audit sessions, configure data capture at native color depth when auditing systems with many concurrent users (such as Citrix XenApp server). When not capturing at native color depth, the DirectAudit daemon has to transform the captured data to its target format which ends up consuming CPU cycles. To automatically set native color depth at installation time, see the documentation.

Grant Audit Installation Rights To Administrator Groups

Centrify Audit Administrators have specific privileges to manage and configure the Server Suite audit configuration. A common mistake is rights are assigned to specific AD accounts vs. AD groups or rights are not delegated at all. When employees leave the company and those AD accounts are disabled, the Audit services becomes inaccessible. To avoid this, Centrify recommends rights over the Audit installation are delegated to AD groups and administrators/auditors are placed into the proper AD groups. This will ensure that all administrators within that security group will have access to configure/modify the audit settings in the event that a specific employee leaves the organization.

Delinea Relationship Best Practices

Monthly Cadence Call with Delinea

Customers can schedule a monthly cadence call with Delinea Account Team, Support, and Engineering to ensure that best practices and customer requirements are consistently being communicated.

Customers should have the contact information of their account team (Account executive, customer success manager, and systems engineer). This account team can help escalate requests internally within Delinea, handle licensing questions and feature requests.

Get Your Annual Delinea Healthcheck

Customers are entitled to annual healthchecks provided by the Delinea account team. Delinea conducts healthchecks in varying scope. A basic healthcheck can be conducted by a Delinea Systems Engineer in a few hours and can provide insight into the usage of Centrify, risks and areas for improvement. Additionally, more advanced healthchecks can be provided in a 3 to 5 day paid engagement with Delinea professional services to delve deep into the environment to provide recommendations for security and operational improvement as well as address identified issues.

Attend Annual Delinea Update Meetings

Customers are entitled to annual update meetings with the Delinea Account and Product Management to understand new feature availability and to submit/track enhancements. Customers often find that Delinea has provided additional capability in a new release that addresses new requirements they are entitled to. Additionally, this helps customers to understand the product roadmap and direction for Delinea and promotes a partnership between Delinea and the customer.

This section describes how to install and use license management tools provided by Delinea to add, remove, monitor, and generate reports about Delinea licenses and license usage. The license management tools described here include the Delinea Licensing Service and the Licensing Report wizard. The license management tools described here allow you to manage licenses for access control, privilege management, and auditing.

This document is intended for administrators who are responsible for adding, removing, monitoring, and reporting on Delinea licenses. Readers should have administrator privileges to use the Delinea Licensing Service, the Licensing Report wizard, and the Access Manager and Audit Manager consoles.

Delinea licenses give you access to the following key features:

- Secure authentication and identity management
- Role-based access rights
- Delegation of authority
- Auditing of activity

These features can be used together or independently, depending on the type of licenses you purchase and the specific requirements of your organization.

Delinea License Management Tools

Delinea provides a set of tools that let you manage Delinea licenses and generate reports about the different types of Delinea licenses you have purchased.

The tool for adding and removing licenses, monitoring license usage, and configuring license usage notification is the *Delinea Licensing Service*. The licensing service works together with Server Suite components to monitor and report usage and activity for all types of Delinea licenses. For more information about using the licensing service, see Managing Licenses with the Licensing Service.

The tool for generating license reports is the *Delinea Licensing Report wizard*. The wizard generates a report summary and detailed system information about the computers where you have Delinea software deployed. Report information is formatted as comma-separated values (CSV) in a text file. The report is intended primarily for use by Delinea Support. If requested, you can send it to Delinea Support for analysis. You can also use the report for your own analysis. For more information about the Licensing Report wizard, see Creating Licensing Reports with the Licensing Report Wizard.

How Licensing Works

Delinea licensing is based on the number of servers and workstations you authorize for authentication and privilege elevation, and for audit and monitoring service. License validation and management are handled through the licensing service, the Access Manager console, and the Audit Manager console.

Delinea license management tools check for license keys when any of the following events occur:

- A manual or automatic refresh operation is performed through the licensing service.
- You start the Access Manager console.
- An agent-managed computer joins a zone.
- You start the Audit Manager console, the Audit Analyzer console, or the auditing session player; or when you rotate to a new auditing database using either PowerShell or the SDK.

Checking licenses for authentication and privilege elevation verifies that there are enough license keys installed for all UNIX and Windows computers with valid accounts in Active Directory.

Checking licenses for audit and monitoring service verifies that there are enough license keys installed for each computer that is connected to an audit and monitoring service collector.

If the number of licensed computers exceeds the total number of licenses you have purchased, a message is displayed with license usage details, and—if applicable—instructions to add license keys.

After you have installed enough license keys to cover all the configured UNIX, Linux, Mac OS X, or Windows computers, the applicable Delinea console will open at startup and allow you to perform all of the normal administrative tasks.

Understanding License Types

Licenses for authentication and privilege elevation (*also referred to as access control and privilege management*) are purchased and installed separately from licenses for *auditing*. You can use the licensing service to install and manage all types of licenses.

Licenses for access control and privilege management are issued based on how a computer is used. For example, a computer can be licensed as a UNIX or Windows workstation or as a standard UNIX or Windows server, or as an application server. The following types of licenses are available:

- **Workstation Licenses (UNIX or Windows)** permit a specific number of UNIX or Windows workstations to be available to Active Directory users. Workstation licenses are intended for computers that are used interactively by one or two concurrent users but that do not host applications accessed by multiple users. There are separate UNIX workstation and Windows workstation licenses.
- **Server Licenses (UNIX or Windows)** permit a specific number of servers to be available to Active Directory users accessing server-based

applications. Server licenses are for computers that are accessed by multiple concurrent users and typically host a specific type of application. There are separate UNIX server and Windows server licenses.

Licenses for audit and monitoring service are issued for each computer that will be connected to an audit and monitoring service collector. Auditing licenses are issued separately for UNIX and Windows computers.

Understanding License Keys

Depending on whether you have purchased software licenses, your license keys might provide limited *evaluation* usage of the software for a specific number of days, or *permanent* access to features for a specific number of computers. If you initially install using an evaluation license key, you must eventually replace that evaluation key with one or more permanent license keys to continue using the software.

Your capacity for enabling access for standard UNIX services or applications is defined by the total of all of the licenses you purchase and install. For example, if you install three valid license keys that each enable 100 workstations for UNIX access, you have a total of 300 workstation login licenses available.

Each license you purchase has a 24-character registration key that specifies:

- The type of license granted by the key.
- The total number of computers that may be enabled under this key's license. If this is an evaluation key, the number of computers is unlimited, but the license count is displayed as zero (0) to indicate no computers are licensed under the evaluation key.
- The time limit for the key. If the license is a permanent license key, the time limit is not applicable. If the license is an evaluation key, the time is set to a duration that is defined in the license key.

Because each license key specifies a set number of computers, it is common to receive multiple license keys. You can provide these license keys when you install Delinea software on a Windows computer or after installation using the licensing service. For information about using the licensing service to add licenses, see Managing Licenses with the Licensing Service.

How License Usage is Counted

How license usage is counted depends on these factors:

- Whether the license is for authentication and privilege elevation (that is, access control and privilege management), or for audit and monitoring service.
- Whether the licensed computer uses the Windows or UNIX operation system.
- Whether the license is a permanent license or an evaluation license.

Using Delinea Licenses in FIPS Environments

Each recently issued Delinea license supports both FIPS compliant and non-FIPS compliant environments. The Licensing Service control panel **DC/DZ Deployment** tab and **DA Deployment** tab show the FIPS status of the selected licensing service host computer.

If you install the licensing service on multiple host computers in the forest (for example, for availability and redundancy reasons), it is highly recommended that all hosts have the same FIPS compliance setting. That is, licensing service hosts should all be FIPS compliant, or should all be non-FIPS compliant.

In most situations, Delinea license management tools are installed by default when you install Server Suite. However, some license management tools also have standalone installers so that you can perform a manual installation, or modify an existing license management tool installation.

Installing the Delinea Licensing Service

The Delinea Licensing Service is installed by default when you install Server Suite. The default installation location is:

C:\Program Files\Centrify\Licensing Service

For details about installing Server Suite, see the *Planning and Deployment Guide* and the *Administrator's Guide for Windows*.

You can choose not to install the licensing service during Server Suite installation by deselecting the **Centrify Licensing Service** check box in the list of components to install when the Delinea Management Services installation wizard executes. For example, if the licensing service is already installed on one computer in the forest and you do not need to install it on other computers, you can deselect it from the list of installed components. However, you should ensure that the licensing service is installed on at least one computer in the forest before deselecting it during installation.

If you do not install the licensing service when you install Server Suite, you can install it separately as described in Performing a Standalone Licensing Service installation.

Performing a Standalone Licensing Service Installation

You can install the Licensing Service separately from Server Suite using the standalone Licensing Service installer.

To perform a standalone installation of the Licensing Service:

1. Log on to the Windows computer and insert the CD or navigate to the directory where you downloaded Delinea files. If the Getting Started page is not automatically displayed, double click the autorun.exe file to start the installation of the Delinea software.
2. On the Getting Started page, click **Centrify Licensing Service**.
3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement, click **I accept the terms in the License Agreement**, then click **Next**.
5. Specify a location for installing licensing service components and click **Next**.
6. Click **Install** to begin the installation.
7. When the installation finishes, click **Finish**.

> **Note**: You can also install the licensing service by executing the following installer, which resides in the Licensing Service folder in the Server Suite installation folder:

Centrify_Licensing_Service-x.x.x-win64.msi

After the licensing service is installed and running, you can use the licensing service control panel to manage Delinea licenses as described in Managing licenses with the Licensing Service.

Modifying a Delinea Licensing Service Installation

You can change, repair, or remove an existing licensing service installation by executing the standalone Delinea Licensing Service installer as described in Performing a Standalone Licensing Service Installation after the licensing service is already installed.

You can select one of the following options:

- **Change.** Select this option to change installation parameters of the current installation.t
- **Repair.** Select this option to remove backup files, re-copy the licensing service program files to the default installation location, and reset registry entries.
- **Remove.** Select this option to remove the licensing service from the computer.

Verifying that the Delinea Licensing Service is Running

To verify that the licensing service is running, open the Services administrative tool on the computer where the licensing service is installed. The "Delinea Licensing Service" entry should have a status of **Started**.

Assigning a License Container to a Zone through Access Manager

If you choose to use more than one license container in the forest, you can assign a specific license container to an individual zone. This option is useful if you want to manage zones independently with each zone using its own set of license keys rather than having all zones use a common pool of licenses. If you assign a specific license container to a zone, however, only the license keys installed in that container can be used for the computers in that zone.

For example, if you create a license container object named ajax.org/Performix Licenses, add a license key for 10 Workstation license to that container, and assigned that container to the Performix Division zone, those 10 workstation licenses are available specifically for the computers you add to the PerformixD ivision zone.

To assign a license container to a zone:

1. Open the Access Manager console.
2. If prompted to connect to a forest, specify a domain controller, and, if needed, the user credentials for connecting to the domain controller, then click **OK**.
3. In the console tree, select **Zones** to display the list of zones.
4. Select a zone and right-click, then click **Properties**.
5. On the General tab, select a specific Licenses container from the list of available **License containers** for the zone to use, then click **OK**.

Installing the Licensing Report Wizard

The Licensing Report wizard is installed by default when you install Server Suite. The default installation location is C:\\Program Files\\Delinea\\Deployment Report.

You can optionally reinstall the wizard manually by executing the Licensing Report wizard standalone installer:

Centrify_Licensing_Report-x.x.x-win64.msi

The standalone installer is located in the DirectManage folder in the Server Suite installation folder.

Modifying a Licensing Report Wizard Installation

You can change, repair, or remove an existing Licensing Report wizard installation by executing the standalone installer:

Centrify_Licensing_Report-x.x.x-win64.msi

The standalone installer is located in the DirectManage folder in the Server Suite installation folder.

You can select one of the following options when you execute the standalone installer in an environment where the wizard is already installed:

- **Change.** Select this option to change installation parameters of the current installation.
- **Repair.** Select this option to remove backup files, re-copy the Licensing Report wizard program files to the default installation location, and reset registry entries.
- **Remove.** Select this option to remove the Licensing Report wizard from the computer.

After the Delinea Licensing Service is installed and running, you can use the Licensing Service control panel to manage Delinea licenses in these ways:

- **Open the Licensing Service control panel.** For more information, see Opening the Licensing Service Control Panel.
- **Start and stop the Centrify Licensing Service, and manually refresh license count information.** For more information, see Starting, Stopping, and Refreshing the Licensing Service.
- **Create license containers for Centrify licenses.** For more information, see Creating License Containers and Adding License Keys.
- **Add and remove licenses for Centrify products.** For more information, see Adding and Removing Centrify License Keys.
- **Monitor license usage for each Centrify product.** For more information, see Monitoring Centrify License Usage.
- **Manually refresh license usage information, and configure how often license usage information is refreshed automatically.** For more information, see Configuring Licensing Service Settings.
- **Configure license usage email notification.** You can configure the list of email recipients, the license usage percentage that triggers notification, whether certain environment details should be omitted from email notification, and outbound email sender details. For more information, see Configuring License Usage Email Notification.
- **View the current licensing service log and configure licensing service logging parameters.** For more information, see Configuring and Viewing Licensing Service Logs.

Opening the Licensing Service control panel

Use the Windows Start menu to open the Licensing Service control panel. If **Centrify Licensing Service Control Panel** is not pinned to the Start menu, use Start menu searching to locate and start the Centrify Licensing Service control panel.

The Licensing Service control panel contains the following tabs:

- **General.** Use this tab to monitor current licensing service status, start and stop the licensing service, and manually refresh license count information as described in Starting, Stopping, and Refreshing the Licensing Service.

  The Current Configuration area displays the name of the Active Directory forest that the licensing service manages, and the name of the computer where the licensing service is running.

  The Monitoring Details area displays the current status of license monitoring, the date of the last license usage update, and license status details. Current status can be one of the following:

  - **Idle.** The licensing service is running, and is available to perform a license usage update.
  - **Busy.** A license usage update is in progress, or the licensing service is starting.
  - **N/A.** The licensing service is not running or is otherwise unavailable.

    An additional field, **Service status**, displays whether the Licensing Service is running or stopped.

- **DC/DZ Deployment.** Use this tab to monitor usage details—including license counts—of authentication and privilege elevation (also known as *access control and privilege management*) licenses. See Monitoring Centrify License Usage for more information about using this tab.
- **DA Deployment.** Use this tab to monitor usage details—including license counts—of audit and monitoring service licenses. See Monitoring Centrify License Usage for more information about using this tab.
- **DC/DZ Licenses.** Use this tab to create license containers, and add or remove licenses for authentication and privilege elevation (also known as *access control and privilege management*). See Creating License Containers and Adding License Keys and Adding and Removing Centrify License Keys for more information about using this tab.
- **DA Licenses.** Use this tab to add or remove licenses for audit and monitoring service. See Adding and Removing Centrify License Keys for more information about using this tab.
- **Settings.** Use this tab to configure how often license usage information is automatically refreshed, and license usage email notification details. Email notification details that you can configure include the list of recipients, SMTP server settings, whether some system information is hidden in notification email, and the license usage threshold that triggers email notification. See Configuring Licensing Service Settings and Configuring License Usage Email Notification for more information about using this tab.
- **Troubleshooting.** Use this tab to manually refresh license usage information and manage licensing service logging. Logging parameters that you can configure include trace level of logged events and log file location. You can also view and save the current log from this tab. For more information, see

Configuring Licensing Service Settings and Configuring and Viewing Licensing Service Logs.

Starting, stopping, and refreshing the licensing service

Use the **General** tab in the Licensing Service control panel to start, stop, and restart the licensing service, and to refresh license count information manually.

To start and stop the licensing service:

1. Open the Licensing Service control panel **General** tab.
2. Click **Start**, or **Stop**, or **Restart**.

Whenever you start or restart the licensing service, license usage information is refreshed. See Monitoring Centrify License Usage for information about viewing license usage information.

Licensing service start and stop information can be logged in the licensing log file depending on the log trace level that is configured. See Configuring and Viewing Licensing Service Logs for information about viewing the licensing log and setting the log trace level.

Refreshing the License Count Manually

In some situations, you might want to refresh the license count information in the licensing service without refreshing all license usage information. When you refresh the license count information as described in this section, license keys are updated in the licensing service, but a complete license usage evaluation is not triggered.

For example, if you add or remove licenses through Access Manager or Audit Manager after the licensing service is installed, you should update the license count in the licensing service as described here so that the license changes that you made through Access Manager or Audit Manager are implemented immediately in the licensing service. If you do not update the license count as described here, license additions or removals that you perform through Access Manager or Audit Manager are not implemented until a regularly scheduled license update task executes.

If you need to refresh all license usage information, including license count, perform the procedure described in Refreshing license usage information.

To refresh the license count manually:

1. Open the Licensing Service control panel **General** tab.
2. Click **Refresh Now**.

Creating License Containers and Adding License Keys

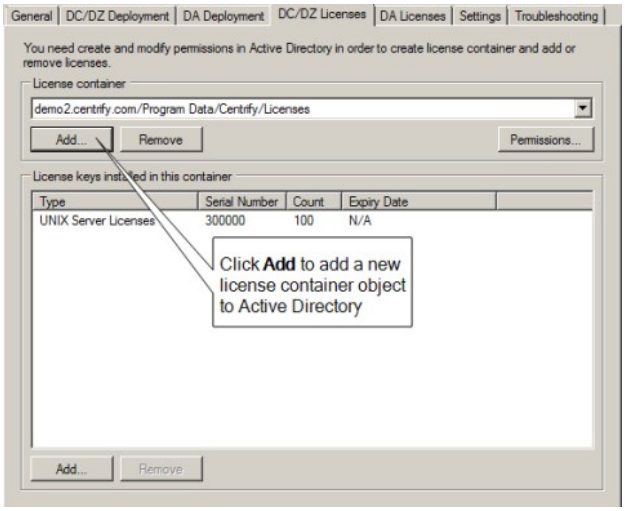
Before you can add Centrify authentication and privilege elevation license keys, you must create a Licenses container object in Active Directory because you must have at least one Licenses container in the forest into which you install license keys. It is also possible to add more License containers to the forest and use those additional containers to control who can use which license keys. For example, you may want to create one license container for application servers and another for workstation licenses. You can then set permissions on the container objects to prevent the workstation administrators from installing the application server license keys and the application server administrators from installing the workstation license keys in their respective containers.

> **Note:** You can also use Access Manager to assign a specific license container to an individual zone as described in Assigning a License Container to a Zone through Access Manager.

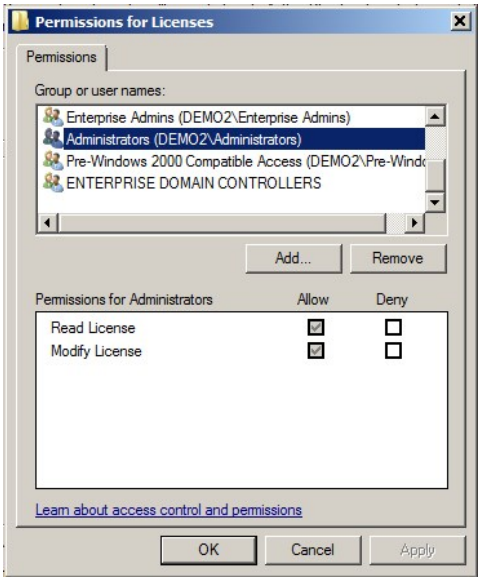
Creating License Containers

To add a new license container object for authentication and privilege elevation licenses:

1. Open the Licensing Service control panel **DC/DZ Licenses** tab.
2. In the License container section, click **Add**.



3. Browse to select a location for the new license container, then click **Create**. Select either container or organizational unit to indicate the type of object to create, and type a name for the new license container object and click **OK**.
4. Click **OK** to close the Browse for Container dialog box.
5. When prompted to confirm the creation of the container object, click **Yes** to add the license container to Active Directory.
6. Click **Permissions** to assign Read License and Modify License permissions to specific users or groups. The users or groups that you give the Modify License permission to can then add license keys to the new license container.



Adding and Removing Centrify License Keys

When you install Centrify software for authentication and privilege elevation, you must provide at least one licensing key that is specific to authentication and privilege elevation. Likewise, when you install audit and monitoring service, you must provide at least one license key that is specific to audit and monitoring service.

> **Note:** If you are using a valid evaluation license key for audit and monitoring service and install a permanent audit and monitoring service license key, the evaluation key is automatically removed from the audit and monitoring service installation. If you are using a valid permanent auditing license key and install an evaluation license key for audit and monitoring service, the evaluation key takes precedence as long as it has not expired.

To add license keys for authentication and privilege elevation:

1. Open the Licensing Service control panel **DC/DZ Licenses** tab.
2. In the **License container** section, select the appropriate License container from the list of available license containers.
3. At the bottom of the dialog box, in the **License keys installed in this container** section, click **Add**.
4. Type or paste the new license key string, then click **OK**.
5. Click **OK**.

To add license keys for audit and monitoring service:

1. Open the Licensing Service control panel **DA Licenses** tab.
2. In the **Installation** section, select the auditing installation for which you are adding a license.
3. Click **Add**.
4. Type or paste the new license key string, then click **OK**.
5. Click **OK**.

To delete a license key that you have previously installed:

1. Open the Licensing Service control panel **DC/DZ Licenses** tab or the **DA Licenses** tab.
2. In the **License keys installed in this container** section, highlight the license to delete, click **Remove**, and then click **OK**.

Monitoring Centrify license Usage

The Licensing Service control panel allows you to monitor how many Centrify licenses in the forest are in use, and how many are available. You can see usage information for access control and privilege management licenses, and for auditing licenses.

> **Note:** To generate a more comprehensive report of license usage, including detailed system information about the computers where you have Centrify software deployed, use the Licensing Report wizard. When you generate a license usage report with the wizard, information is formatted as comma-separated values (CSV) in a text file. See Creating Licensing Reports with the Licensing Report Wizard for more information.

To see usage information for authentication and privilege elevation licenses:

1. Open the Licensing Service control panel **DC/DZ Deployment** tab.

   The license usage information that is displayed when you first open the tab is from the last time you refreshed the information in the **DC/DZ Deployment** dialog box. If you restarted the Licensing Service or refreshed license usage information from the **Troubleshooting** tab, the latest license usage information is not displayed in the **DC/DZ Deployment** dialog box until you perform Step 2 and Step 3.

2. In the **Source** field, select the licensing service host computer for which to report license usage information.

   If you select **Last Updated Centrify Licensing Service**, the licensing service host computer containing the latest valid licensing information is selected.

3. Click **Refresh**.

   Information displayed in the dialog box is based on the most recent refreshing of license usage information, which is generated by a scheduled refresh, manual refresh, or starting the Licensing Service. That is, clicking **Refresh** here does not instruct the licensing service to check current license usage.

4. Optional: To display current license usage information:

   1. Perform a manual refresh from the **Troubleshooting** tab as described in Configuring Licensing Service Settings.
   2. Return to the **DC/DZ Deployment** tab and click **Refresh**.

To see usage information for audit and monitoring service licenses:

1. Open the Licensing Service control panel **DA Deployment** tab.

   The license usage information that is displayed is for all available audit and monitoring service installations (that is, global capacity and usage are shown).

2. In the **Source** field, select the licensing service host computer for which to report license usage information.

If you select **Last Updated Centrify Licensing Service**, the licensing service host computer containing the latest valid licensing information is selected.

3. In the **Installation** field, select an audit and monitoring service installation whose licensing information you want to see.
4. Click **Refresh**.

   Information displayed in the dialog box is based on the most recent refreshing of license usage information, which is generated by a scheduled refresh, manual refresh, or starting the licensing service. That is, clicking **Refresh** here does not instruct the licensing service to check current license usage.
5. Optional: To display current license usage information:
   1. Perform a manual refresh from the **Troubleshooting** tab as described in Configuring Licensing Service Settings.
   2. Return to the **DA Deployment** tab and click **Refresh**.

Configuring Licensing Service Settings

Use the **Settings** tab in the Licensing Service control panel to refresh license usage information and configure licensing email notification details.

Refreshing license usage information

You can use the licensing service to refresh license usage information in these ways:

- Start or restart the licensing service as described in Starting, Stopping, and Refreshing the Licensing Service.
- Configure an automatic refresh interval as described in this section.
- Perform a manual refresh of license count and usage as described in this section.

To configure an automatic refresh interval:

1. Open the Licensing Service control panel **Settings** tab.
2. In the Update Schedule section, click **Configure**.
3. Specify a start time and recurrence pattern (interval), and click **OK**.

To manually refresh license count and usage information:

1. Open the Licensing Service control panel **Troubleshooting** tab.
2. Click **Refresh Now**.

   Note: Manually refreshing license count and usage as described here updates both license count and license usage. If you want to update just the license count (for example, due to performance considerations), perform the procedure described in Refreshing the License Count Manually.

Configuring License Usage Email Notification

Email is sent to a list of recipients when a license is deployed. Routine notices are sent if license usage is below 90% (or a different threshold if you specify one). Warnings are sent if usage thresholds are exceeded. The default thresholds that trigger email warnings are 90%, 100%, and 120%.

Typical email notification appears as follows, and includes an attachment of the latest report generated by the Licensing Report wizard (not shown here):

`From: "Delinea Licensing Service" <licensingservice@domain.com>

To: "System Administrator" <sysadmin@domain.com>

Subject: Notice from Centrify: You are deploying 17% of your DirectControl/DirectAuthorize UNIX licenses, 20% of your DirectAuthorize Windows licenses, and 30% of your total DirectAudit licenses.

Date: Monday, October 31, 2016 12:00 AM

DirectControl/DirectAuthorize UNIX usage: 13 out of 75
DirectAuthorize Windows usage: 5 out of 25
Total DirectAudit usage: 15 out of 50

=== DirectControl/DirectAuthorize Usage Summary ===
UNIX Server: 5 used, 50 licensed
UNIX Workstation: 5 used, 25 licensed
UNIX without license type: 3 used
Windows Server: 2 used, 25 licensed
Windows Workstation: 3 used, 0 licensed
DirectControl/DirectAuthorize Evaluation: 0 licensed
DirectControl UNIX - Express: 0 used
Mac systems: 0 used
zLinux systems: 0 used

=== DirectAudit Usage Summary ===
UNIX Server: 5 used, 25 licensed
UNIX Workstation: 5 used, 0 licensed
UNIX without license type: 0 used
Windows Server: 2 used, 25 licensed
Windows Workstation: 3 used, 0 licensed
DirectAudit Evaluation: 0 licensed
zLinux systems: 0 used`

You can configure email notification details such as the list of recipients, SMTP server settings, whether to omit some system information from the report attached to the notification email, and one license usage threshold that replaces the default 90% threshold.

To configure license usage email notification:

1. Open the Licensing Service control panel **Settings** tab.
2. In the Notification section, click **Configure**.
3. To add or remove a user on the email recipient list:
    - To remove the user from the list of recipients, highlight the user name and click **Remove**.
    - To add a user to the list of recipients, click **Add** in the Recipients section. Type the email address and display name for the user and click **OK**.
4. To configure SMTP server settings:
    - Type the display name and email address of the user who is the email sender.
    - Type the host name or IP address of the outbound email server and specify a port number.
    - If the outbound email server requires secured password authentication for login, select **This server requires authentication** and type the name and password of a user with login permission. If the outbound email server does not require secured password authentication for login, leave this check box unchecked.
    - To optionally test the SMTP configuration, click **Test** and specify a recipient for the test email message.
5. To specify that a warning is sent when license usage exceeds a threshold other than 90%, select **Warn when the usage reached** and specify a threshold value from 1% to 89%. The threshold that you specify is used instead of the default 90% threshold.
6. The license usage report generated by the Licensing Report wizard is included as an attachment in email notification. To omit host, zone, and installation names from the attachment, select **Hide host, zone and installation names from the attached licensing report**.
7. Click **OK**.

    Note: Depending on how a recipient's email filters are configured, notification email might be redirected to a recipient's junk or spam folder. If a recipient is not receiving email as expected, check the recipient's junk or spam folder. If necessary, modify the recipient's email filters so that notification email is not directed to the junk or spam folder.

    Note: By default, if the license usage report is 10 MB or larger, the service sends the report as a .zip file instead of a plain text attachment. You can add the following registry setting to configure the size for which the service attaches the report as .zip file: HKLM\SOFTWARE\Centrify\Licensing Service\ReportNotificationCompressionThreshold as a DWORD registry setting. Enter the value as the number of MB for the file size threshold. For example, a value of 3 will make sure that licensing reports that are 3 MB or larger are sent as zip files.

Configuring and Viewing Licensing Service Logs

Detailed licensing service log files are created while the licensing service is running, and are saved in the following default location:

C:\\Program Files\\Common Files\\Centrify Shared\\Logs

In addition to viewing the current log file, you can edit the current file, configure which events are stored in log files (that is, the logging level) and specify the location where log files are stored.

To view and edit the current log file:

1. Open the Licensing Service control panel **Troubleshooting** tab.
2. To save the current log file without first opening it for viewing, click **Save As**, navigate to a folder location, specify a name for the log file, and click **Save**.
3. To open the log file for viewing and editing, click **View Log**. While the file is open for viewing, use the **File** and **Edit** menus to save the file to a name and location of your choice, select parts of the file for copying, or clear the file.
4. To save a zipped version of the log file to a location of your choice, click **Export Diagnostics Data**, specify a folder location, and click **OK**.

To configure licensing service event logging level:

1. Open the Licensing Service control panel **Troubleshooting** tab.
2. Click **Options**.
3. In the Log Settings dialog box, select one of the following categories of events to log:
    - No log messages
    - Error messages
    - Warning messages
    - Informational messages
    - All messages
4. Click **Apply**, then click **OK**.

To specify a different licensing service log file folder:

1. Open the Licensing Service control panel **Troubleshooting** tab.
2. Click **Options**.
3. In the **Log folder path** field, specify or browse to the folder where you want to store licensing service log files.
4. Click **OK**.

The Licensing Report wizard collects information about the Delinea software you have deployed, including how many licenses you have installed, where they are used, where they are inactive, and the number of licenses that remain available in the forest. Information is reported for audit and monitoring service, authentication and privilege elevation licenses.

The wizard is installed by default when you install Server Suite on a Windows computer. You can also download and install the wizard separately using a standalone setup program as described in Installing the Licensing Report Wizard.

Note: Depending on the version of your installed Delinea software, the following nomenclature caveats could apply:

- Licenses for authentication and privilege elevation might be shown as *DirectControl and DirectAuthorize* licenses in examples and command output.
- Licenses for audit and monitoring service might be shown as *DirectAudit* licenses in examples and command output.

Permissions Required to Generate a Licensing Report

You must have the following privileges to generate a licensing report:

Centrify Express for UNIX/Linux	A user account in a domain that is trusted by other domains in the forest, or an account that can search and read information from all of the domains in the forest.
Authentication & Privilege	A user account in a domain that is trusted by other domains in the forest, or an account that can search and read information from all of the domains in the forest.
Audit & Monitor	A user account with the permissions required for Authentication & Privilege plus the “Manage License” or “View” permission for the audit and monitoring service installation.

You can specify different user accounts when you run the wizard, if needed. For example, you might need to use different accounts to collect information about one or more audit and monitoring service installations.

Information Required to Produce the Licensing Report

Before you run the wizard, verify that you have the following information available:

Folder location	The Licensing Report wizard can be installed with Server Suite or downloaded and installed as a separate executable file. If you download the file from the Centrify website, it is saved by default in your Downloads folder. This folder is usually located on the drive where Windows is installed in a folder with your user name. For example, the path to the file might be similar to this: C:\Users\your_name\Downloads
Domain controller	Centrify products are licensed for the entire forest. Therefore, the wizard must be able to connect to a domain controller that can access Active Directory information for the entire forest. You are prompted to specify the domain controller and credentials for connecting to the domain when you start the wizard. Alternatively, you can specify the name of a domain that is trusted to access other domains in the forest.
User credentials for authentication and privilege elevation (*DirectControl and DirectAuthorize*)	The wizard must be able to read license and deployment information from the domain controller that has access to all domains in the forest. By default, your logon account credentials are used to connect to the domain controller. If your logon account does not have the List Objects permission to access the domain controller, you can specify a different user name and password when prompted to specify the account credentials for connecting to the domain.

User credentials for audit and monitoring service (*DirectAudit*)	The licensing report wizard attempts to retrieve usage information from Active Directory before attempting to retrieve information from audit store databases. The wizard attempts to connect to audit store databases only if the information is not found in Active Directory. If the wizard needs to connect to an audit store database, user credentials and the "Manage License" or "View" permission for each installation are required. By default, the same user credentials are used to get all deployment information. However, if the user account does not have the "Manage License" or "View" permission, you can specify a different user name and password for audit installations when prompted.
Report output configuration	You need to specify the file name and folder location for the licensing report generated by the wizard. You must also decide whether the report output should show or hide information about the computers where you have deployed Centrify software. If you choose to hide zone, computer, and installation names, the information will be replaced with a one-way hash of the text to prevent the computers from being identified in the report. If you choose this option, you will not be able to review and validate license information for specific computers. The report output is saved as comma-separated values (CSV) in a text file.

Preparing to run the Licensing Report wizard

To keep the report output concise, Delinea recommends that you check for and remove orphaned computer accounts and decommissioned computers before generating a licensing report.

To check for and remove orphaned and decommissioned computers:

1. Open Active Directory Users and Computers and delete the computer objects associated with the decommissioned computers.
2. Open Access Manager.
3. Right-click **Centrify Access Manager** in the navigation pane, then select **Analyze**.
4. Select **Orphan zone data objects and invalid data links**, then click **Next**.
5. Click **Finish**.
6. Select Analysis Results to check whether any orphan information was found.

   If there are deleted computer objects with an orphan zone profile listed in the Analysis Results, select the issue, right-click, then select **Remove orphan profile**.

Running the Licensing Report Wizard

After you have installed the Licensing Report wizard as part of Server Suite or as a separate standalone installation package, you can run the wizard to generate a licensing report. You can start the wizard from within Access Manager or by navigating to it from the Start menu.

To run the wizard from within Access Manager:

1. Open Access Manager.
2. Right-click **Centrify Access Manager**, then select either **Centrify Licensing Report (DirectControl)** or **Centrify Licensing Report (DirectControl & DirectAudit)**.

   If you select DirectControl, the licensing report utility automatically checks the current forest for authentication and privilege elevation information using your current account credentials. If you need to specify different credentials or check audit and monitoring service licenses, select **Centrify Licensing Report (DirectControl & DirectAudit)** or from the Start menu select **Centrify Licensing Report**.
3. Click **Next** to accept the default domain controller and your current credentials to retrieve deployment information.

If necessary, you can specify a different domain controller and select the option to specify a different user if your current account does not have permissions to retrieve deployment information, then click **Next**.

4. If the Audit Management service is not running, or if DirectManage Audit version 2015.1 or earlier is installed, or if you do not have the necessary audit and monitoring service permissions, you are prompted to specify credentials to retrieve audit installation information.

   If you see this prompt, click **Next** to use your current credentials to retrieve audit installation information.

   If necessary, you can select the option to specify a different user, then specify a different user name and password if your current account does not have the "Manage Licenses" or "View" permission for the audit Installation, then click **Next**.

   If you do not see this prompt, go to Step 6 and continue from there.

5. If the Audit Management service is not running, or if DirectManage Audit version 2015.1 or earlier is installed, or if you do not have the necessary audit permissions, you are prompted to specify whether your current credentials can be used to retrieve audit and monitoring service installation information.

   If you see this prompt, click **Next** if the credentials specified in Step 4 can retrieve information for all of the audit installations listed.

   If necessary, you can select an audit and monitoring service installation and click **Change Credentials** to specify a different user name and password for connecting to a specific installation, then click **Next**.

   If you do not see this prompt, go to Step 6 and continue from there.

6. Specify the name and folder location for the licensing report and whether to hide host, zone, and installation names in the report output, then click **Next**.

   - By default, the licensing report output is located in your Documents folder with a name in the format of Centrify_Licensing_Report_yyyymmdd.txt, where yyyymmdd is the year, month, and date indicating when you are generating the report. If a report of the same name already exists in that location, a version number suffix is added to the default report name.
   - Select the option to **Hide host, zone, and installation names from the report** to keep this information private. The wizard will generate random strings to replace host, zone, and installation names in the report output. Note that selecting this option does not obfuscate the Active Directory forest name. The forest name is required to send the report output to Delinea. All other names included in the report can be replaced with random strings.

7. Review the output location and file name, then click **Next** to generate the report.

8. To preview the report before saving it or sending it to Delinea, click **Preview Report**.

   To open the report for editing or to save it as a different file name, leave the **Open the output report** option selected and click **Exit**.

   To send the report output directly to the Delineay Support portal, click ***Send to Delinea**.

9. Click **OK** to acknowledge that the report will be sent and continue.

   You will be given a reference number for communicating with support about the report and prompted to log in using your Delinea account user name and password. After logging in, click **Continue** to display details about your report.

Running the utility as a separate package

You can access the shortcut for the licensing report executable directly from the Start menu. If **Licensing Report** is not pinned to the Start menu, use Start menu searching to locate and start the licensing report utility. After you open the utility, the steps for generating the report are the same as the steps in the previous section. Follow the instructions in the wizard to generate the report output.

Running the utility from the command line.

As an alternative to running the licensing report utility as a wizard, you can use the command-line interface to run the wizard in a Command Prompt window. To use the command-line interface for the utility, navigate to the directory where the CentrifyDeploymentReport.exe file is located (the default location is C:\Program Files\Centrify\Deployment Report). Open a Command Prompt window, and execute the command using the following syntax:

```
CentrifyDeploymentReport.exe [/standardmode]    [/server=server] [/plaindata] [/silent /output=filepath  [/force]] [/help] [/?]
```

You can use the following options with the utility:

/standardmode	Run CentrifyDeploymentReport.exe with standard edition support only.
/server=server	Specify the name of a domain controller in the forest for which you want to run the report.
/plaindata	Include host, zone, and installation names in the report. By default, host, zone, and installation names are not included in the report.
/silent	Run CentrifyDeploymentReport.exe in silent mode. You can use this option when generating the report for Server Suite standard edition using the /standardmode option, or without the /standardmode option to generate an report that includes audit and monitoring service information.
/output=filepath	Specify output file path and file name of the licensing report. You can use this option only when you are using the /silent option.
/force	Force the generation of a new licensing report even if the output file specified already exists. You can use this option only when you are using the /silent option.
/help, /?	Display command syntax and usage information.

Reviewing the licensing report output

The Licensing Report wizard generates a report formatted as a set of comma separated values (CSV) in a text file. The report contains two main sections:

- The first section contains summary information about the counted computers where you have Delinea software deployed.
- The second section contains detailed information about the computers where you have Delinea software deployed, including separate areas for counted and uncounted computers. If a computer is uncounted, a comment explains the reason why it is uncounted.

> **Note:** See How Computers are Counted for Licensing Reports for more information about which computers in the forest are counted, and how their licenses count against the total number of available licenses.

The first and second report sections are separated from each other as follows:

Just before the end of Section 1, a checksum is included to validate the authenticity of the report. For example:

```
Checksum,1,"frACfH0SRjhEDxPFU5ZAbfoZ5ISMKm1ZFqssWG79V4Wr3QC4Fp1wneQG03U26C+lU0608J5PdrV2vuH0nMJLxcdi6cV4nerrZPhmhlIf7MU="
```

Editing the checksum or any other part of Section 1 invalidates the report. If you make any changes in this section, you will need to generate a new report.

You should also note that the last lines in the report are a report identifier string and the version number of the Licensing Report wizard that generated the report. For example, you might see lines similar to this at the end of the report:

```
Report ID,"8OY5i6p0LZtePMYTAg0PqcImIZA="
Version,"5.4.0.118"
```

You should not modify or delete the report identifier or the version number.

How Computers are Counted for Licensing Reports

To generate a report, the licensing report software first determines which computers in the forest are validly using Centrify software. These “valid usage” computers are considered “counted” computers. Licenses for counted computers are subtracted from the total number of available permanent workstation or server licenses, and their licensing summary information is reported in Section 1 of the licensing report.

Counted Computer Scenarios

A computer is counted if the following scenarios are true:

- The computer's zone status is **Auto Zone** or **Zoned**. This scenario applies only to computers where authentication and privilege elevation features are installed.
- The computer has a status of **Active**.

Uncounted Computer Scenarios

Uncounted computers are included (together with counted computers) in Section 2 of the licensing report, but are not shown in Section 1 because their licenses are not subtracted from the total number of available licenses.

A computer is uncounted if any one of the following scenarios is true:

- The computer's zone status is **Express** or **Zoneless**. This scenario applies only to computers where authentication and privilege elevation features are installed.
- The computer is using an authentication and privilege elevation license, and is joined to the **Null Zone**. Note that computers using audit and monitoring service licenses are counted even if they are joined to the null zone.
- The computer has a status of **Inactive**.
- The computer is using an authentication and privilege elevation license, and is **Orphaned** (the computer profile exists in the zone but the corresponding Active Directory computer object has been removed). Note that computers using audit and monitoring service licenses are counted even if they are orphaned.
- The computer has a **Duplicated** audit and monitoring service license (the audit and monitoring service agent was migrated from one installation to another, and the time stamp of the agent from the earlier installation has not expired).
- The computer has an **Unknown logon time** (the computer has never joined the domain).

License Type Information for Managed and Audited Computers

If you have authentication and privilege elevation features or audit and monitoring service features deployed, the summary in report Section 1 includes information about the type of Delinea license in use on each computer.

License type can be one of the following values:

- **Server**
- **Workstation**
- **None** (The license type cannot be determined from the Active Directory object, as is the case when the computer is orphaned, or the agent is from a release earlier than 2015.1.)

See Understanding License Types for more information about license types.

Zone information for managed computers

If you have authentication and privilege elevation features deployed, report Section 1 includes zone mode information in the "DirectControl/DirectAuthorize Agent Type" string shown in Example 2: Zone Mode and Number of Agents.

> **Note:** Zone mode applies only to computers where authentication and privilege elevation features are installed. Zone mode does not apply to computers using audit and monitoring service licenses.

Depending on the nature of your deployment, the zone mode information displays one of the following values:

- **Auto Zone** if the computer is in a Centrify Auto Zone.

  You cannot use Centrify rights and roles on computers joined to an Auto Zone.

  If the Zone mode for a computer is Auto Zone, the computer is included in the authentication service (DirectControl) license count.

- **Zoned** if the computer is in a standard Centrify zone.

  All authentication and privilege elevation features are supported for computers in Centrify zones on most platforms. However, the Centrify Agent for Mac OS does not support Centrify rights and roles.

  If the Zone mode for a computer is Zoned, the computer is included in the authentication service (DirectControl) license count.

- **Express** if the computer has a Centrify Express agent installed.

  Computers with a Centrify Express agent have limited functionality. For example, you cannot apply group policies or use Centrify rights and roles on

computers with the Centrify Express agent.

If the Zone mode for a computer is Express, the computer is not included in the authentication service (DirectControl) license count.

- **Zoneless** if a computer has the Centrify Agent installed but is not connected to a zone.

  This agent type is primarily for computers that use Centrify MFA for Windows login authentication.

  If the Zone mode for a computer is Zoneless, the computer is not included in the authentication service (DirectControl) license count.

- **Null Zone** if a computer is joined to the null zone.

  If the Zone mode for a computer is Null Zone, the computer is not included in the authentication service (DirectControl) license count.

See Example 2: Zone Mode and Number of Agents for details about how zone mode information is displayed in the report.

Status information for managed and audited computers

If you have authentication and privilege elevation features or audit and monitoring service features deployed, the counted/uncounted information in report Section 2 indicates the status of the computer as **Active** or **Inactive**:

- **Active** if the computer has been used for authentication and privilege elevation, or for audit and monitoring service, within 45 days prior to the date that the report was run.

  Computers with an active status are included in the license count.

  For authentication and privilege elevation licenses, the time stamp of the managed computer logon to the domain controller is monitored if the functional level of the domain controller is Windows Server 2003 or later. The licensing report uses the time stamp of the LastLogonTimestamp attribute to determine whether there has been logon activity within 45 days prior to the date that the report was run.

  For audit and monitoring service licenses, the licensing report uses the most recent time that the managed computer has communicated with a collector to determine whether there has been auditing activity within 45 days prior to the date that the report was run.

- **Inactive** if the computer has not been used for authentication and privilege elevation, or for audit and monitoring service, within 45 days prior to the date that the report was run.

  Inactive computers are not included in the license count.

See Example 6: Counted Identity and Privilege Elevation Computers for an example of computer status information.

Remarks for Managed and Audited Computers

If you have authentication and privilege elevation features or audit and monitoring service features deployed, the counted/uncounted information in report Section 2 includes remarks about the following computers:

- Uncounted computers with authentication and privilege elevation licenses.
- Counted and uncounted computers with audit and monitoring service licenses.

Remarks provide additional information about why a computer is uncounted, and other significant information to be aware of. See Example 7: Counted Audit and Monitoring Service Computers and Example 8: Uncounted Computers of All License Types for examples of remarks strings.

The Remarks string can have the following values:

- **Duplicated** if an audit and monitoring service agent was migrated from one installation to another (such as during an upgrade), and the time stamp of the agent from the earlier installation has not expired.

  Computers with duplicated licenses are not counted. That is, the license is only counted once.

- **Excluded due to null zone** if the computer was not counted because it is joined to the null zone.

  Computers with authentication and privilege elevation features (DirectControl and DirectAuthorize licenses) are not counted if they are joined to the null zone.

  Computers with audit and monitoring service features (DirectAudit licenses) are included in the auditing license count even if they are joined to the null

zone.

- **Excluded due to zoneless mode** if the computer has a zone mode of Zoneless (that is, the computer has the Centrify Agent installed but is not connected to a zone).
- **Excluded due to express mode** if the computer has a Centrify Express agent installed.
- **Inactive** if the computer has a status of Inactive as described in Status Information for Managed and Audited Computers.
- **None** if no additional information is required.
- **Orphaned** if the computer profile exists in the zone but the corresponding Active Directory computer object has been removed.

  Orphaned computers with authentication and privilege elevation features (DirectControl and DirectAuthorize licenses) are not included in the license count. You can use Access Manager to delete orphan profiles as described in Preparing to run the Licensing Report wizard.

  Orphaned computers with audit and monitoring service features (DirectAudit licenses) are included in the auditing license count if the computer has communicated with a collector within 45 days prior to the date that the report was run.
- **Unknown logon time** if the computer has never joined the domain. This situation typically occurs when you use Access Manager to prepare a UNIX computer prior to joining the computer to the domain. Computers with an unknown logon time are not included in the license count.
- **Vault-based systems** are your Windows, UNIX, and/or network devices that are managed by and audited by the Server Suite.

Evaluation Licenses for Managed and Audited Computers

If a computer has a valid evaluation license, the detailed section of the licensing report (Section 2) indicates the licensing status as Evaluation (Valid). In the case of valid evaluation licenses, the summary section of the report (Section 1) might show the "Available" licenses as a negative number. You can ignore negative available licenses if you have valid unlimited evaluation licenses. However, if the licensing status indicates an expired evaluation license, you should remove the expired evaluation license key.

Status Information for Zoneless Computers

If a computer has the Centrify Agent installed but is not connected to a zone, its agent type is listed as *Zoneless* in the licensing report. This agent type is primarily for computers that use Centrify MFA for Windows login authentication.

Examples

This section contains examples of a hypothetical licensing report.

Depending on the version of your installed Centrify software, the following nomenclature caveats could apply:

- Licenses for authentication and privilege elevation are shown as *DirectControl and DirectAuthorize* licenses.
- Licenses for audit and monitoring service are shown as *DirectAudit* licenses.

> **Note:** "Join Time" is the point in time when the computer joined to the zone. Because Join Time is a feature introduced in Release 2020, if the agent is older than Release 2020, the Join Time displays as "Unknown."

Example 1: Agent, License Type and Count

The following example shows the first portion of report Section 1, containing summary system information. Colored lines indicate how entries in each section relate to each other. Different line colors are for readability only.

Note that this example shows the agent (DirectControl, DirectAudit, and/or DirectAuthorize), the license type (UNIX, Windows, or combined UNIX and Windows), licenses found, licenses used, and licenses available.

Depending on what types of Centrify licenses you have in your environment, your own licensing report could contain fewer entries than the example shown here.

Example 2: Zone Mode and Number of Agents

The following example shows the next portion of report Section 1, displaying the deployment quantities for each type of DirectControl/DirectAuthorize agent based on zone mode. This example shows the layout of this section rather than example data. Depending on what types of Centrify licenses you have in your environment, your own licensing report might not use all of the layout entries shown here.

DirectControl/DirectAuthorize Agent Type,Deployed Agents

"Zoned Server Windows",#
"Zoned Server Mac",#
"Zoned Server zLinux",#
"Zoned Workstation",#
"Zoned Workstation Windows",#
"Zoned Workstation Mac",#
"Zoned Workstation zLinux",#

```
"Zoned (Workstation or Server)",#
"Zoned Mac (Workstation or Server)",#
"Zoned zLinux (Workstation or Server)",#
"Auto Zone Server",#
"Auto Zone Server Mac",#
"Auto Zone Server zLinux",#
"Auto Zone Workstation",#
"Auto Zone Workstation Mac",#
"Auto Zone Workstation zLinux",#
"Auto Zone (Workstation or Server)",#
"Auto Zone Mac (Workstation or Server)",#
"Auto Zone zLinux (Workstation or Server)",#
"Null Zone Server",#
"Null Zone Server Mac",#
"Null Zone Server zLinux",#
"Null Zone Workstation",#
"Null Zone Workstation Mac",#
"Null Zone Workstation zLinux",#
"Null Zone (Workstation or Server)",#
"Null Zone Mac (Workstation or Server)",#
"Null Zone zLinux (Workstation or Server)",#
"Zoneless Server Windows",#
"Zoneless Workstation Windows",#
"Express",#
"Express Mac",#
"Express zLinux",#
```

Example 3: Zone Names and Deployment Details

The following example shows the next portion of report Section 1. Information for DirectControl/DirectAuthorize deployments is sorted by zone. Information

for DirectAudit deployments is sorted by agent type and by installation. This example shows the layout of this section rather than example data. Depending on what types of Centrify licenses you have in your environment, your own licensing report might not use all of the layout entries shown here.

```
Number of Zones,#
------------------------------------------------------------
DirectControl/DirectAuthorize Zone,Deployed Agents,Location
"Whi8ewrOe/",#,"Z0QApzH7++"
"4eS3i2Cccq",#,"cWepjBPNZ5"
...
```

```
DirectAudit Agent Type,Deployed Agents
-----------------------------------------
"Server UNIX/Linux",#
"Server zLinux",#
"Server Windows",#
"Workstation UNIX/Linux",#
"Workstation zLinux",#
"Workstation Windows",#
"UNIX/Linux (Workstation or Server)",#
"AuditedMachine",#
```

```
DirectAudit Installation,Version,Status,Deployed Agents
------------------------------------------------------------
"kiXnFshYnq","2.0 or later","OK",#
"Nt+njcALLE","2.0 or later","OK",#
"2y8grBYVHP","1.3 or earlier","OK",#
...
```

Example 4: License Detail Summaries

The following example shows the final portion of report Section 1. It displays a summary for authentication and privilege elevation (DirectControl/DirectAuthorize) licenses, and a summary for audit and monitoring service (DirectAudit) licenses.

The Count string in this section is especially useful to check the total number of installed licenses. Other details include license keys, serial numbers, and expiration dates.

If the Shared string displays Yes, the license key is being shared by more than one audit and monitoring service installation.

At the end of Section 1 is a checksum that validates the authenticity of the report. Do not edit the checksum or any other content preceding it before sending the report to Centrify for analysis. This example shows the layout of this section rather than example data. Depending on what types of Centrify licenses you have in your environment, your own licensing report might have more or fewer entries than the layout shown here.

```
License Report for DirectControl/DirectAuthorizeAgent
----------------------------------------------------------------------
Type,License Key,Count,Serial Number,Expiry Date

Evaluation (Valid),XXXXXXXX-XXXXXXXX-XXXXXXX,#,None,"d MM yyyy"

UNIX Server,XXXXXXXX-XXXXXXXX-XXXXXXX,#,######,"None"

UNIX Workstation,XXXXXXXX-XXXXXXXX-XXXXXXX,#,######,"None"
...
```

```
License Report for DirectAudit
--------------------------------
Agent Type,License Key,Count,Serial Number,Expiry Date,DirectAudit Installation,Shared

Evaluation (Valid),XXXXXXXXXXXXXXXXXXXXXXXX,#,0,"d MM yyyy","GmZ9u0po4M",No

UNIX Server,XXXXXXXXXXXXXXXXXXXXXXXX,#,######,"None","g/qJ26pGdk",No
...
Checksum,1,"frACfH0SRjhEDxPFU5ZAbfoZ5lSMKm1ZFqssWG79V4Wr3QC4Fp1wneQG03U26C+lU0608J5PdrV2vuH0nMJLxcdi6cV4nerrZPhmhllf7MU="
==== END OF REPORT SUMMARY. DO NOT MODIFY ANYTHING ABOVE THIS LINE =====
```

Example 5: Counted and Uncounted Computers

The following example shows the first portion of report Section 2, containing information about whether a computer is or is not counted in usage calculations. The counted summary section is a copy of the summary from Section 1 for reference.

```
===== END OF REPORT SUMMARY. DO NOT MODIFY ANYTHING ABOVE THIS LINE =====
------------------------------------------------------------------------
```

```
Counted Usage Summary

Agent Type,Licenses Found,Counted Usage,Licenses Available
DirectControl/DirectAuthorize Server - UNIX,#,#,#
DirectControl/DirectAuthorize Server - Windows,#,#,#
DirectControl/DirectAuthorize Workstation - UNIX,#,#,#
DirectControl/DirectAuthorize Workstation - Windows,#,#,#
DirectControl/DirectAuthorize Server - ALL (summary),#,#,#
DirectControl/DirectAuthorize Workstation - ALL (summary),#,#,#
DirectAudit Server - UNIX,#,#,#
DirectAudit Server - Windows,#,#,#
DirectAudit Workstation - UNIX,#,#,#
DirectAudit Workstation - Windows,#,#,#
DirectAudit Server - ALL (summary),#,#,#
DirectAudit Workstation - ALL (summary),#,#,#
```

```
Uncounted DirectControl/DirectAuthorize Usage,#
Uncounted DirectAudit Usage,#
```

Example 6: Counted Identity and Privilege Elevation Computers

The following example shows the next portion of report Section 2, containing information about counted computers where authentication and privilege elevation (DirectControl and DirectAuthorize) features are used. Information includes the system name, the timestamp of the most recent Active Directory update, OS and agent versions, zone mode (see Zone Information for Managed Computers), status (see Status Information for Managed and Audited Computers), current zone, and license type (see License Type Information for Managed and Audited Computers.

Information about these counted computers is collected and reported in Section 1 of the report, as shown in Example 1: Agent, License Type and Count through Example 4: License Detail Summaries.

This example shows the layout of this section and example data. Depending on what types of Centrify licenses you have in your environment, your own licensing report might not use all of the layout entries shown here.

```
System Report of Counted Usage for DirectControl/DirectAuthorize
---------------------------------
Number of Systems - Counted DirectControl/DirectAuthorize Server - ALL (summary),#
Number of Systems - Counted DirectControl/DirectAuthorize Workstation - ALL (summary),#
Number of Systems - Counted DirectControl/DirectAuthorize - Grand Total,#
System,Last Computer AD Timestamp,OS,OS Version,Agent Version,Zone Mode,Status,Current Zone,License Type,Join Time,Postal Address
"gqxYNiU4cB","dd MMM yyyy HH:mm:ss zzz","CentOS","6.2","5.3.1-394","Zoned,Active,"28eQ86egjQ"",Server,"dd MMM yyyy HH:mm:ss zzz","JoinTime:xxxxxxxxxx;Key:value"

"ob6eV5JqUa","dd MMM yyyy HH:mm:ss zzz","Windows 7 Enterprise","6.1 (7601)","3.3.0-161","Zoned",Active,"dkXp8d1Bhp",Workstation,"dd MMM yyyy HH:mm:ss zzz","JoinTime:xxxxxxxxxx;Key:value"

"8m3DWH/ixr","dd MMM yyyy HH:mm:ss zzz","Red Hat Enterprise Linux","7.2","5.3.1-339","Zoned",Active,"KiWfCdeOlm",Server,"dd MMM yyyy HH:mm:ss zzz","JoinTime:xxxxxxxxxx;Key:value"

"CGAeslRVqB","dd MMM yyyy HH:mm:ss zzz","Scientific Linux","6.0","5.3.1-382","Auto Zoned",Active,"nQ+FzJZiGl",Workstation,"dd MMM yyyy HH:mm:ss zzz","JoinTime:xxxxxxxxxx;Key:value"

...
```

Example 7: Counted Audit and Monitoring Service Computers

The following example shows the next portion of report Section 2, containing information about counted computers where audit and monitoring service (DirectAudit) features are used. Information includes the system name, the timestamp of the most recent communication with a collector, OS and agent versions, status (see Status Information for Managed and Audited Computers), license type (see License Type Information for Managed and Audited Computers), and remarks (see Remarks for Managed and Audited Computers).

Vault-based systems are your Windows, UNIX, and/or network devices that are managed by and audited by the Server Suite.

Note that the remarks string for one computer states "Orphaned," but the computer is still counted because audited computers are counted even when they are orphaned (unlike DirectControl/DirectAuthorize computers).

Information about these counted computers is collected and reported in Section 1 of the report, as shown in Example 1: Agent, License Type and Count through Example 4: License Detail Summaries.

This example shows the layout of this section and example data. Depending on what types of Centrify licenses you have in your environment, your own licensing report might not use all of the layout entries shown here:

```
System Report of Counted Usage for DirectAudit
-----------------------------------------------------------
Number of Systems - Counted DirectAudit Server - ALL (summary),#
```

```
Number of Systems - Counted DirectAudit Workstation - ALL (summary),#

Number of Systems - Counted DirectAudit - Grand Total,#

System,Last Connection,OS,Agent Version,Status,DirectAudit Installation,License Type,Join Time,Remarks,Postal Address,Audit Type,Advanced Monitoring,Role Based Audit,Session Reviews,Deployment Type

"ORiVwQ8knZ","dd MMM yyyy HH:mm:ss zzz","Windows",3.3.1-391,Active,"Rc6xacwgT4",Workstation,"dd MMM yyyy HH:mm:ss zzz","None","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Enabled","Unknown","No","Local"

"vRGdi5XdYd","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",3.3.0-161,Active,"xiAKQayRc0",None,"dd MMM yyyy HH:mm:ss zzz","Orphaned","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Enabled","Unknown","No","Local"

"DzFls6sbyG","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",3.3.0-161,Active,"zP0V7QsMEG",Workstation,"dd MMM yyyy HH:mm:ss zzz","None","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Disabled","Unknown","Yes","Local"

"o3oXhLNF8Q","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",Unknown (2.0 or later),Active,"0Z9opN8afT",None,"dd MMM yyyy HH:mm:ss zzz","Vault-based system","JoinTime:xxxxxxxxxx;Key:value","Gateway Based","Enabled","Unknown","No","Cloud"

"LNFVToyG16","dd MMM yyyy HH:mm:ss zzz","Windows",Unknown (2.0 or later),Active,"MOYdPMpxJk",Server,"dd MMM yyyy HH:mm:ss zzz","Vault-based system","JoinTime:xxxxxxxxxx;Key:value","Gateway Based","Enabled","Unknown","Yes","Cloud"

"xnSXEWuUPM","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",Unknown (1.3 or earlier),Active,"qZ3cKtElMz",None,"dd MMM yyyy HH:mm:ss zzz","None","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Enabled","Unknown","No","Local"

...
```

Example 8: Uncounted computers of all license types

The following example shows the last portion of report Section 2, containing information about uncounted computers (considered "invalid usage" computers) where authentication and privilege elevation (DirectControl, DirectAuthorize) features, and audit and monitoring service (DirectAudit) features might be deployed.

DirectControl and DirectAuthorize information includes the system name, the timestamp of the most recent Active Directory update, OS and agent versions, zone mode (see Zone Information for Managed Computers), current zone, license type (see License Type Information for Managed and Audited Computers), and remarks (see Remarks for Managed and Audited Computers).

Note that the Remarks string shows that DirectControl and DirectAuthorize computers were not counted because the were inactive, had an unknown logon time, were orphaned, or were joined to the null zone.

DirectAudit information includes the system name, the timestamp of the most recent Active Directory update, OS and agent versions, license type (see License Type Information for Managed and Audited Computers), and remarks (see Remarks for Managed and Audited Computers).

Note that the Remarks string shows that DirectAudit computers were not counted because they were inactive or duplicated.

> **Note:** The Active Directory attribute of postalAddress is repurposed to store the agent join date as well as additional information for UNIX/Linux systems. The postalAddress field does not store data such as zip codes. Because postalAddress is a multi-value string, the format displayed in the report is like "JoinTime:xxxxx;LicenseType:xxxx".

This example shows the layout of this section and example data. Depending on what types of Centrify licenses you have in your environment, your own

licensing report might not use all of the layout entries shown here:

System Report of Uncounted Usage for DirectControl/DirectAuthorize

---

Number of Systems - Uncounted DirectControl/DirectAuthorize Usage,#

System,Last Computer AD Timestamp,OS,OS Version,Agent Version,Zone Mode,Current Zone,License Type,Join Time,Remarks,Postal Address

"jWSNWAVqSE","dd MMM yyyy HH:mm:ss zzz","Red Hat Enterprise Linux","6.2","5.3.0-127","Zoned","7TKshsizX+",Workstation,"dd MMM yyyy HH:mm:ss zzz","Inactive","JoinTime:xxxxxxxxxx;Key:value"

"BQ74st+6DY","dd MMM yyyy HH:mm:ss zzz","Windows Server 2012 Standard","6.2 (9200)","3.3.0-161","Zoned","IXcmEK0/GA",Server,"dd MMM yyyy HH:mm:ss zzz","Inactive","JoinTime:xxxxxxxxxx;Key:value"

"iRuAXmdP8R","None","Unknown","Unknown","None","Zoned","DcXp7iwKdT",None,"dd MMM yyyy HH:mm:ss zzz","Unknown logon time","JoinTime:xxxxxxxxxx;Key:value"

"a52fpZViDQ","None","Unknown","Unknown","None","Unknown","MWbqK6NgEg",None,"Orphaned","JoinTime:xxxxxxxxxx;Key:value"

"didvGnjncv","dd MMM yyyy HH:mm:ss zzz","Red Hat Enterprise Linux","6.4","5.2.2-186","Null","Wrhxp83dy6",None,"dd MMM yyyy HH:mm:ss zzz","Excluded due to null zone","JoinTime:xxxxxxxxxx;Key:value"

"vvoJLHbwEz","dd MMM yyyy HH:mm:ss zzz","Windows 7 Enterprise","6.1 (7601)","3.4.0-100","Zoneless","None",None,"dd MMM yyyy HH:mm:ss zzz","Excluded due to zoneless mode","JoinTime:xxxxxxxxxx;Key:value"

"qNi04XTWUH","dd MMM yyyy HH:mm:ss zzz","SUSE Linux","12.0","5.3.1-369","Express","None",None,"dd MMM yyyy HH:mm:ss zzz","Excluded due to express mode","JoinTime:xxxxxxxxxx;Key:value"

Number of Systems - Uncounted DirectAudit Usage,#

System,Last Connection,OS,Agent Version,DirectAudit Installation,License Type,Join Time,Remarks,Postal Address,Audit Type,Advanced Monitoring,Role Based Audit,Session Reviews,Deployment Type

"Fw/yiZD26M","dd MMM yyyy HH:mm:ss zzz","Windows",3.3.0-129,"JLcHyjSGTO",Workstation,"dd MMM yyyy HH:mm:ss zzz","Inactive","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Enabled","Unknown","No","Local"

"ouS7+ZC6qt","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",3.2.2-246,"hcZs4xQ80F",None,"dd MMM yyyy HH:mm:ss zzz","Inactive","JoinTime:xxxxxxxxxx;Key:value","Agent Based","Disabled","Unknown","No","GP"

```
"vsJgzxtg6A","dd MMM yyyy HH:mm:ss zzz","UNIX/Linux",3.3.0-161,"FjuO9D63g2",Workstation,"dd MMM yyyy HH:mm:ss zzz","Duplicated","JoinTime:xxxxxxxxxx;Key:value","Gateway Based","None","Unknown","Yes","Cloud"

...
```

Example 9: List of Zones and Special Profiles

The following examples show the additional sections of the report that show detailed information about zones and special user profiles.

The Zone report lists detailed information about each zone. Below is a sample.

```
Zone Report

----------------------

CN,Type,Parent Zone,Tenant ID,Tenant URL,Is ZPA provisiong,Is agentless client supported,Number of user profiles,Number of group profiles,Number of local user profiles,Number of local group profiles,Number of local Windows user profiles,Number of local Windows group profiles,Number of cross domain users,Number of cross forest users,Number of NIS maps,Number of roles,Number of PAM rights,Number of Command rights,Number of Windows Desktop rights,Number of Windows Application rights,Number of Windows Network Access rights,Number of role assignments,Number of computer roles,Number of roles with "MFA required",Number of roles with "audit if possible",Number of roles with "audit required",Number of roles with "audit not required"

keJg5xag6R,$CimsZoneVersion7,FbiyaZP2IT,AAQ0527,http://xxxxxxxxxx,

Yes,No,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#

...
```

In the next section of the licensing report lists the Special AD User Profiles Defined Report. This report lists out any special user profiles (SCP) that have been defined; profiles are detected based on pattern matching such as "oracle", "hadoop", "hdfs", "hive", and so forth.

You can configure the pattern match my modifying the registry value (value name: "SpecialUserProfilePattern"; type: multi string). Each row stands for a defined name pattern (from all zones).

If there is no name pattern configured, this section is not displayed in the report.

```
Special AD User Profiles defined Report

----------------------------------------

"oracle": #

"hadoop": #

"hdfs": #

"hiv$": #

...
```

The next section of the licensing report is the Special Local User Profiles Defined Report, which is very similar to the previous section. This report lists out any special local user profiles (SCP) that have been defined; profiles are detected based on pattern matching, such as "oracle", "hadoop", "hdfs", "hive", and so forth.

You can configure the pattern match my modifying the registry value (value name: "SpecialUserProfilePattern"; type: multi string). Each row stands for a defined name pattern (from all zones).

If there is no name pattern configured, this section is not displayed in the report.

```
Special Local User Profiles defined Report
------------------------------------------

"oracle": #
"hadoop": #
"hdfs": #
"hiv$": #
...
```

The guide provides the following information:

Preparing For An Upgrade

This chapter provides an overview of the upgrade process and a summary of the compatibility requirements between the core components of Server Suite software. You should review the information in this chapter before upgrading any components on the computers where Delinea software is installed.

Upgrading the Operating System

Upgrading the operating system (OS) on a managed computer can make major changes to the configuration files and utilities installed on it, In many cases, operating system upgrades and operating system patches can changes affect the behavior of Delinea software. If the behavior of Delinea software is modified because of an operating system upgrade, it is possible for users to be locked out and unable to access to computer resources. To prevent this from happening, Delinea recommends that you first remove any Delinea packages you have installed before upgrading the operating system, then reinstall the packages after the operating system upgrade has been completed and the computer has been verified to be operating normally.

You should note that removing Delinea software prior to applying operating system patches or upgrading the operating system is not required in most cases. However, because operating system changes can affect authentication and authorization services, it is considered a best practice to ensure the upgrade does not interrupt services for any users.

Upgrading Computers Accessed by Multiple Users

In most cases, you can upgrade Delinea software on computers that are accessed by multiple users without entering single-user mode. However, upgrading authentication, authorization, and auditing services on a computer can potentially prevent users from logging on or using computer resources. If possible, you should perform upgrades when other users who might access the computer are logged off, then reboot the computer after completing the upgrade.

You should note that having all users logged off and rebooting the computer after an upgrade are not required steps, but are best practices to ensure the upgrade does not interrupt services for any users. In most cases, users who are already logged on are not affected by the upgrade. However, users who attempt to log on while files are being replaced during the upgrade process might be temporarily locked out of the managed computer you are upgrading.

Compatibility Between Versions of Delinea Software

In most cases, newer versions of Delinea software releases are backward-compatible with previous versions, enabling you to mix and match components from different versions and upgrade components over time when it is convenient to do so. However, there are some limitations to take into account when mixing and matching versions, and these limitations might influence which components you upgrade and how quickly you upgrade from one version to another.

In most organizations, the agents you install on managed computers are upgraded on a staggered schedule while administrative tools are upgraded at a set time to take advantage of new features.

To ensure flexibility of the upgrade process:

- Agents are always backward-compatible with older versions of the administrative console.

  However, using an older version of the administrative console with a newer agent limits the features and functionality available. If you are using an administrative console from version 2.x to manage zones, agents from version 4.x and 5.x must use the --compat option to join 2.x-compatible zones.

- Agents are always forward-compatible with the administrative console for one version.

  You can upgrade the administrative console without upgrading agents at the same time. However, there are limitations to features and functionality when using older agents with an upgraded console. For example, agents from version 4.x cannot be included in hierarchical zones. In addition, some features require an upgrade. For example, if you want to use the Delinea Agent for Windows for access control and privilege management, you must either upgrade or remove the Delinea auditing service for Windows.

- Group policies are not guaranteed to be compatible with different agent and administrative console versions.

  New group policies cannot be enforced on computers with an agent from a previous version of Delinea software. If a group policy is applied to a computer that has an older version of the agent, the policy is ignored. You should only apply group policies that are supported in both the agent and administrative console versions you are using.

Finding Upgrade Packages

You can find Server Suite and agent packages for all supported operating systems on the Delinea Customer Download Center. From the Customer Download Center, you can choose to download individual agent packages one at a time or download an archive that includes agents for all operating systems at once.

At a minimum, you should download the Delinea Agent Installer and the ADCheck Diagnostic Tool. You can then use the install.sh shell script interactively or with the centrify-suite.cfg configuration file to install and enable features on the computers you want to upgrade. Delinea recommends that you use the install.sh shell script to install or upgrade all Delinea packages on managed computers, especially if you have multiple Delinea packages installed that you wish to upgrade. The install.sh installation script performs a thorough set of pre-installation and post-installation steps to ensure a successful installation or upgrade with minimal disruption to your environment.

Alternatively, you can use the native package manager for your operating system to upgrade the components you have installed. If you want to use a native package manager, see Using a native package manager on Linux computers for Linux computers or Using a native package manager on UNIX computers for UNIX computers.

Disabling Command-line Auditing

If you have auditing enabled on a computer you are upgrading, you should check whether auditing is configured for individual commands or all user activity. If you have enabled auditing for specific commands, you should temporarily disable auditing on the managed computer before upgrading, then restart the auditing of individual commands after completing the upgrade. If you are auditing all user activity on a managed computer, you do not need to stop the auditing service. There will be a brief interruption while files are replaced, then auditing will continue without requiring you to manually restart it.

Upgrading Delinea Management Services on Windows Computers

This section describes how to upgrade Authentication & Privilege and Audit & Monitoring administrative components on Windows computers. It includes a more detailed discussion about compatibility between components.

> **Note**: In releases before 2017.2, you installed the DirectManage Access and DirectManage Audit sets of components. Those component areas have been renamed and are now called Authentication & Privilege and Audit & Monitoring,respectively.

What Should You Upgrade First?

You are not required to upgrade Delinea software components in any particular order. Depending on where you have components installed and how they are distributed, you might update components used for auditing before updating components for access control and privilege management. Alternatively, you might update one set of agents immediately, followed by one administrative console, then update other components at a later time.

Although there's no technical requirement to upgrade components in a specific order, most organizations upgrade one or more administrative consoles and components that might require changes to a database first—for example, Access Manager if upgrading access control and privilege management—then deploy upgraded agent software after upgrading all of other components.

Similarly, if you upgrading the auditing infrastructure, you might upgrade Audit Manager, the management database, and the audit store before upgrading collectors and agents.

Updating Administrative Components

As noted in General compatibility between versions of Delinea software, most organization upgrade the administrative consoles at a set time, often as part of planned maintenance, then upgrade agents opportunistically over a period of time. It is common, therefore, to have a mix of components from different versions of Delinea software within certain limits.

To help you plan for the upgrade, you should identify which versions of different components you currently have installed and which components will require an upgrade.

Depending on whether you are upgrading Authentication & Privilege, Audit & Monitoring, or both feature sets, you might have different compatibility requirements.

Access Control and Privilege Management Compatibility

You can upgrade to Server Suite—with Access Manager, version 5.1.x or later—to manage zones and agents (adclient) from version 3.x, 4.x, or 5.x. If you have agents from version 2.x, you must manage them using a console from version 4.x or earlier. If you use an older version of the console, you cannot take advantage of any features or enhancements introduced in newer versions of the console. If you upgrade to the latest release, you can continue to manage all of your currently deployed agents but must upgrade those agents to take full advantage of any new features.

You must upgrade UNIX, Linux, or Mac agents to 5.0 or later to use hierarchical zones. If you have zones from a previous release of Delinea software, you can use admigrate to convert those zones to hierarchical zones.

To manage Windows computers with Access Manager, the Delinea Agent for Windows must be version 3.0 or later.

Auditing Infrastructure Compatibility

You can upgrade to Server Suite—with Audit Manager, Audit Analyzer, and Collector service version 3.1.x or later—to manage auditing on UNIX, Linux, and Windows computers from version 2.x or 3.x. If you have agents from version 1.x, you must manage them using a console from version 1.x.

You must update the collector service to version 3.x to receive audit data from Windows computers with 3.x Windows agents.

Because the auditing infrastructure is a multi-tiered architecture that collects information to be preserved, reviewed, and archived, Delinea recommends a more formal upgrade process than for other components. This is especially true for larger organizations that collect a great deal of audit data. If you are upgrading the auditing infrastructure, therefore, see Upgrading the auditing infrastructure for more detailed information about the process to follow.

Access Control and Privilege Management Compatibility

You can upgrade to Server Suite—with Access Manager, version 5.1.x or later—to manage zones and agents (adclient) from version 3.x, 4.x, or 5.x. If you have agents from version 2.x, you must manage them using a console from version 4.x or earlier. If you use an older version of the console, you cannot take advantage of any features or enhancements introduced in newer versions of the console. If you upgrade to the latest release, you can continue to manage all of your currently deployed agents but must upgrade those agents to take full advantage of any new features.

You must upgrade UNIX, Linux, or Mac agents to 5.0 or later to use hierarchical zones. If you have zones from a previous release of Delinea software, you can use admigrate to convert those zones to hierarchical zones.

To manage Windows computers with Access Manager, the Delinea Agent for Windows must be version 3.0 or later.

Auditing Infrastructure Compatibility

You can upgrade to Server Suite—with Audit Manager, Audit Analyzer, and Collector service version 3.1.x or later—to manage auditing on UNIX, Linux, and Windows computers from version 2.x or 3.x. If you have agents from version 1.x, you must manage them using a console from version 1.x.

You must update the collector service to version 3.x to receive audit data from Windows computers with 3.x Windows agents.

Because the auditing infrastructure is a multi-tiered architecture that collects information to be preserved, reviewed, and archived, Delinea recommends a more formal upgrade process than for other components. This is especially true for larger organizations that collect a great deal of audit data. If you are upgrading the auditing infrastructure, therefore, see Upgrading the auditing infrastructure for more detailed information about the process to follow.

Upgrading Components Interactively

You can upgrade components on any Windows computer interactively by clicking the links on the Server Suite Getting Started page. If the setup program detects components are installed, you have the option to update, modify, or remove those components. You can then follow the prompts displayed to review the components to be updated and complete the upgrade.

If the setup program detects components are installed, you are prompted to confirm that you want to continue with the upgrade. You can then follow the prompts displayed to review the components to be updated and complete the upgrade.

Upgrading Auditing Components Silently on Windows

If you want to perform a "silent" or unattended installation of the Delinea auditing components, you can do so by specifying the appropriate command line options and Microsoft Windows Installer (MSI) file to deploy. You can also use an unattended installation to automate the installation or upgrade on remote computers if you use a software distribution product, such as Microsoft System Center Configuration Manager (SCCM), to deploy software packages.

If you have the physical CD or ISO image for Delinea software, you can find the Microsoft Windows Installer (MSI) files for auditing components in subdirectories under the DirectAudit folder.

Before running the Microsoft Windows Installer (MSI) for any component, you should verify the computers where you plan to install meet the prerequisites described in the *Auditing Administrator's Guide*.

To install the auditing components silently:

1. Open a Command prompt window or prepare a software distribution package for deployment on remote computers.

   For information about preparing to deploy software on remote computers, see the documentation for the specific software distribution product you are using. For example, if you are using Microsoft System Center Configuration Manager (SCCM), see the Configuration Manager documentation.

2. Select the appropriate package for the auditing component you want to upgrade.

   For example, locate the following file to install the audit management server on 64-bit operating systems:

   Centrify DirectAudit Audit Management Server64.msi

3. Run the installer with no user interface and specify the package for the auditing component you want to upgrade.

   For example, to upgrade an agent on 64-bit operating systems, run the following command:

   msiexec /qn /i "Delinea Agent for Windows64.msi"

Upgrading Auditing Infrastructure

This chapter describes the recommended steps for upgrading auditing-related components to ensure you can continue auditing activity throughout the upgrade process. Keep in mind that upgrading the auditing infrastructure might require updates to the existing database, but, in most cases, should not require any computers to be shutdown or restarted to complete the upgrade.

Delinea

Why Are There Formal Steps for Upgrading an Audit Installation

In most organizations that deploy auditing, the auditing infrastructure—the installation—consists of components on multiple computers that must be able to communicate with each other to collect, transfer, and store information about user and computer activity. This multi-tiered architecture might be widely distributed and might include hundreds or thousands of computers that must be monitored. Upgrading all of those computers without interrupting ongoing auditing service requires a formal upgrade process that allows computers from different versions to continue communicating for a period of time.

Upgrading Auditing Components in a Specific Order

Because the upgrade process is expected to take a period of time—the length of time depends on the size and complexity of your installation—there are specific rules about the configurations supported and the order in which you should upgrade auditing components. To ensure auditing continues uninterrupted during the upgrade period, you should upgrade audit installation components in the following order:

1. Audit store databases
2. Management server databases
3. Consoles and collectors and the management server service
4. Agents

By following this upgrade order, you can ensure components can continue to communicate while you upgrade the rest of the audit installation. For example, an upgraded audit store can continue to receive audit date from collectors and respond to requests from the management server and consoles that have not be updated.

Be sure to upgrade all of your audit store databases before upgrading other components. You can upgrade the database without upgrading other components from a Command window by running the following command:

```
setup.exe /database
```

Unsupported Configurations

If you upgrade auditing components in a different sequence than the one described in Upgrading auditing components in a specific order, you might end up with an unsupported configuration that requires you to upgrade the remaining components immediately or suspend auditing of user activity until you can complete the upgrade.

You might encounter this situation if you upgrade the Audit Manager and Audit Analyzer consoles or a collector before upgrading the management and audit store databases.

Updating Auditing-Related Databases

If an upgrade requires an update to the database, you are prompted to run the database maintenance wizard and to select the databases to upgrade. If the wizard can connect to the databases selected and the database upgrade is successful, no further action is required.

You can upgrade audit store databases and the management database interactively using the Database Maintenance Wizard or by running the following command:

```
setup.exe /database
```

Upgrading the auditing databases, however, requires specific Windows and database permissions. Before attempting to upgrade the database, verify that you have a user account that meets the following requirements:

- The Windows account you use to update the database with the Database Maintenance Wizard must be an Active Directory domain user and a local administrator on computer where you are running the setup.exe program.
- Your Windows or SQL login account must be either a member of sysadmin fixed server role or a member of db_owner database role on each of the database instances being upgraded. If the account is a member of db_owner database role, you must also have the EXTERNAL ACCESS ASSEMBLY permission on each of the database servers hosting the management database and audit store databases.

You can use the following SQL statement to grant the EXTERNAL ACCESS ASSEMBLY permission to a specific user:

GRANT EXTERNAL ACCESS ASSEMBLY TO [DOMAIN\user]

For example, to grant this permission to the account john@acme.com, you might execute the following SQL statement:

GRANT EXTERNAL ACCESS ASSEMBLY TO [ACME\john]

Updating Agents Out of Sequence

The recommended upgrade steps suggest that you to update deployed agents last. However, upgrading the agent is much simpler than upgrading the audit store or management database. which might require a database administrator to be involved. In most cases, it is safe to update the agent at any point in the upgrade process. If there are restrictions that would prevent a new agent from using an older collector, those restrictions are documented in the release notes.

Restarting Computer After Agent Upgrade

If a computer has both Delinea Privilege Elevation Service and Delinea Audit & Monitoring Service enabled, you must restart the computer after upgrading the agent. If a computer only has Delinea Audit & Monitoring Service, there's no requirement to restart.

Delinea

Best Practices for Upgrading Large Audit Installations

When upgrading the Delinea Audit & Monitoring Service environment to a newer version, always follow this order for minimal service disruption:

1. Upgrade the audit databases.
2. Upgrade the audit collectors.
3. Upgrade the agents.
4. Upgrade remaining components (such as the consoles, PowerShell cmdlets, SDK, Audit Management server and so forth).
5. Delinea recommends using the Database Maintenance Wizard to "Generate the SQL scripts" for upgrading the databases rather than letting the Database Maintenance wizard perform an in-process upgrade.
6. When performing a database upgrade by manually running the scripts, follow this order for minimal service disruption:
    - Upgrade the "Active" audit store database(s).
    - Upgrade the remaining audit store databases.
    - Upgrade the audit management database.
7. To upgrade the audit databases, typically the sysadmin rights on the database server are needed. This is because some of the operations performed during the database upgrade (such as marking an assembly with EXTERNAL ACCESS) requires permissions that are typically only assigned to the users with sysadmin rights. If the database environment is hardened and the user cannot run the upgrade scripts as a sysadmin, here's a minimum set of permissions that the user must have in order to upgrade the databases:
    - User must have db_owner rights
    - User must have the EXTERNAL ACCESS ASSEMBLY rights
8. If the user is unsure whether the user has necessary rights to run the database upgrade or not, we recommend that the user generate the SQL scripts for database upgrade and hand them over to the database administrator for execution.
    - The audit database upgrade scripts are idempotent, which means accidentally running them multiple times will not cause any harm.
9. The database upgrade scripts sometimes log warning messages (for example, for exceeding key length). These warnings can be safely ignored. However, if any errors are received while running the database upgrade scripts, please notify Centrify support. This information is also available in KB-32276.

Upgrading Managed Computers

This chapter describes how to update Delinea software on managed Linux and UNIX computers. You can also upgrade Delinea software on Mac OS X computers using the install.sh shell script in a Terminal application or by downloading, unpacking, and running the latest Mac OS X installer. For more information about upgrading Delinea software on Mac OS X computers, see the **Administrator's Guide for Mac**.

> **Note**: When you upgrade agents installed on Linux computers, the upgrade process does not automatically enable desktop auditing on those systems. For information about enabling desktop auditing on supported Linux systems, see the **Auditing Administrator's Guide**.

Configuring install.sh to Run Without User Interaction

You can use the install.sh shell script to upgrade computers silently without user interaction. When you run install.sh without user interaction, you have the same upgrade options that you have when using install.sh interactively. When using install.sh without user interaction, however, you specify the type of upgrade on the command line and in a configuration file.

--std-suite upgrades authentication and privilege elevation features. Any other Delinea packages you have installed are unchanged as long as they are compatible with the version being upgraded.

--ent-suite upgrades authentication, privilege elevation, Delinea-enabled OpenSSH, and auditing features. Any other Delinea packages you have installed are unchanged as long as they are compatible with the version being upgraded.

In both cases, you can customize the upgrade by modifying the default centrify-suite.cfg configuration file. With the default centrify-suite.cfg configuration file, the install.sh script upgrades the Delinea Agent access control and privilege management features or the Delinea access control, privilege management, and auditing features.

If you have already installed OpenSSH before you upgrade, either upgrade option also upgrades the OpenSSH packages.

All other packages available are left unchanged. For more detailed information about configuring a silent upgrade using the configuration file, see “Setting the parameters in a custom configuration file for the installation script” and the details for the INSTALL parameter in the Planning and Deployment Guide.

Using the install.sh Shell Script to Update Packages

The Delinea Agent installation script, install.sh, is a shell script that you can run interactively or configure to run silently on any supported UNIX, Linux, or Mac OS X computer.

You can use the install.sh shell script to upgrade any installed Delinea software except Delinea sudo. If you have the Delinea sudo package, you can upgrade the package before or after you upgrade the Delinea Agent and other packages.

To use the install.sh script interactively:

1. Unzip and extract the contents of the file you downloaded from the Delinea Customer Download Center. For example:

   gunzip centrify-infrastructure-services-*<release>*-platform-arch.tgz
   tar -xvf centrify-infrastructure-services-*<release>*-platform-arch.tar

2. Run the install.sh script to start the update on the local computer's operating environment. For example:

   ./install.sh

   The installer checks that it is possible to update Delinea software on the local computer. For example, it will check that the computer is a supported platform and that any required patches are installed. For more information about the ADCheck diagnostic tool, see the *Planning and Deployment Guide*.

3. Specify the type of upgrade you want to perform.

   - (E) option: This option upgrades Server Suite access control, privilege management, secure shell, and auditing features. Any other Delinea packages you have installed are unchanged as long as they are compatible with the version being upgraded.
   - (S) option: This option upgrades Server Suite access control, privilege management, and secure shell (Centrify-enabled OpenSSH) features. Any other Delinea packages you have installed are unchanged as long as they are compatible with the version being upgraded.
   - Custom (C) option: This option allows you to select the Delinea packages located in the current directory and choose whether to erase (E), update (U), reinstall (R), keep unchanged (K) each package. If there is a package available for which there is no corresponding version already installed, you can choose to install the package.
   - (X) option: This option installs or upgrades the access control and privilege management components as unlicensed Delinea Express components.

     If you want to install or upgrade additional packages such as the Delinea Network Information Service (adnisd) or the Delinea LDAP proxy service, you should use the custom install option and select the packages to install.

Using a Native Package Manager on Linux Computers

When you upgrade using the install.sh shell script, the script manages all dependencies and compatibility issues for you. If you want to upgrade Delinea software packages using the native package manager, you should first determine whether there are any compatibility issues or dependencies between the packages you have installed. For details about specific version compatibility requirements and upgrade scenarios, see Compatibility for Additional Packages.

As of version 5.4.0, the core Delinea Agent bundle consists of four packages that must always be upgraded to the same version simultaneously: Delinea, CentrifyDC-openssl, centrify-openldap, and CentrifyDC-curl. When fixes and patches are released, you can update individual packages of the core bundle, as long as the version is the same version as the other core packages.

After you have determined whether you have any version dependencies, you can use the native package manager to upgrade packages simultaneously. You can also use the native package manager to remove old packages individually or remove all packages simultaneously.

If you want to install or upgrade software packages using common native package installers, such as the Red Hat or Debian package manager, you should note that the software packages are signed with a GNU Privacy Guard (GPG) key. You need to import the key to verify the package authenticity before installing or upgrading the package. To import the key, download the RPM-GPG-KEY-centrify file from the Delinea Download Center then run the appropriate command for the package manager. For example:

```
rpm --import RPM-GPG-KEY-centrify
```

If you are not using a native package manager, you can use any other installation program you have available for the local operating environment. For example, if you use another program, such as SMIT, YAST, APT, or YUM to install and manage software packages, you can use that program to install Server Suite software packages.

Upgrading Packages on a Linux Computer

You do not need to stop any running Delinea process to perform the upgrade. While you do not usually need to restart Delinea processes or reboot your computer after upgrade, you may need to restart other processes that depend on PAM or NSS modules. Rebooting the computer after upgrade is recommended as a best practice.

It is best to install all Delinea packages simultaneously, if you are upgrading individual packages, however, you might see warnings from the package manager about package dependencies or version conflicts. If you see that a dependency is generated because of a package you have yet to upgrade, it is safe to ignore the warning.

Fresh Installation Using RPM

If you are performing a fresh installation on a Linux computer that supports the Red Hat Package Manager (rpm), you can install the packages individually. For example, to install the Delinea Audit & Monitoring Service package you would enter commands similar to the following:

```
rpm -i CentrifyDA-5.4.0*-platform.arch.*rpm
```

The platform and architecture you specify in the file name on the command line should identify the specific operating system you are using, for example Centrify-5.4.0-rhel4.x86_64.rpm or centrify-5.4.0-suse10.ia64.rpm. After the package manager updates the packages installed, you can optionally restart Delinea processes or reboot the computer.

You can verify the Delinea packages that were installed using the following command:

```
rpm -qa CentrifyDC-\*
```

Upgrading Existing Packages Using RPM

If you are upgrading an existing installation of an agent package on a Linux computer that supports the Red Hat Package Manager (rpm), you should add all of the packages you want to upgrade to a directory of your choice, and issue a single command similar to this:

```
rpm -Uhv my_dir/\*.rpm
```

Where my_dir is a directory that you specify.

Fresh Installation Using the Debian Package Manager

On a Debian, Ubuntu, or Linux MINT computer, the order that you install the core package depends on whether you are performing a fresh installation or upgrading an existing installation. Any Delinea packages other than the core packages can be listed after the core bundle in any order.

For example, to perform a fresh installation of the core authentication service package, you would enter commands similar to the following:

```
dpkg -i ./centrifydc-openssl-5.4.0-platform-arch.deb
./centrifydc-openldap-5.4.0-platform-arch.deb
./centrifydc-curl-5.4.0-platform-arch.deb
./centrifydc-5.4.0-platform-arch.deb
```

Upgrading Packages Using the Debian Package Manager

If you are upgrading an existing installation, the order of the core packages is different than that in a fresh installation. Delinea packages other than the core packages can be listed after the core bundle in any order.

For example, if you were updating all of the Delinea Agents, you would enter commands similar to the following, noting that the packages in **bold** are the core agent packages, and must be entered in the order below:

```
dpkg -i --force-confnew --force-confmiss
\--ignore-depends=centrifydc-nis
\--ignore-depends=centrifydc-ldapproxy
\--ignore-depends=centrifyda./centrifydc
__./centrifydc-5.4.0-***platform-arch***.deb__
**./centrifydc-openssl-5.4.0-***platform-arch***.deb**
**./centrifydc-openldap-5.4.0-***platform-arch***.deb**
**./centrifydc-curl-5.4.0-** *platform-arch***.deb**
./centrifydc-ldapproxy-5.4.0-*platform-arch*.deb
./centrifydc-nis-5.4.0-*platform-arch*.deb
./centrifyda-3.4.0-*platform-arch*.deb
```

> **Note**: If you are upgrading the core agent package from version 5.4.0 or later, to any later version, you must include Delinea in the list of packages to ignore. If you do not have centrifydc-nis, centrify-ldapproxy, or centrifyda installed, the --ignore-depends command for those packages is not necessary.

The platform and architecture you specify on the file name in the command line should identify the specific operating system you are using, for example centrifydc-5.4.0-deb7-i386.deb. After the package manager updates the packages installed, you can optionally restart Delinea processes or reboot the computer.

You can verify the Delinea packages that were upgraded using the following command:

```
dpkg -s CentrifyDC-\*
```

Using a Native Package Manager on UNIX Computers

When you upgrade using the Delinea install.sh shell script, the script manages all dependencies and compatibility issues for you. If you want to upgrade Delinea software packages using the native package manager, you should first determine whether there are any compatibility issues or dependencies between the packages you have installed. You can then upgrade packages individually or simultaneously. For details about specific version compatibility requirements and upgrade scenarios, see Compatibility for additional packages.

After you have determined whether you have any version dependencies, you can use the native package manager to upgrade all packages simultaneously. You can also use the native package manager to remove old packages individually or remove all packages simultaneously.

Upgrading Packages Individually on a UNIX Computer

With the exception of Solaris computers, you do not need to stop any running Delinea process to perform an upgrade on UNIX machines. On Solaris computers, you should stop all Delinea processes before upgrading. You should note that while rebooting the computer or restarting agent services after an upgrade is not required for Delinea processes in most cases, you may need to reboot the computer or restart any processes that rely on PAM or NSS modules after you complete the upgrade to ensure that the upgraded binaries and libraries are being run. Rebooting the computer after upgrade is recommended as a best practice.

To upgrade Delinea software using the native package manager, follow these basic steps:

- Stop all Delinea processes running on Solaris computers.

  For example:

  /usr/share/centrifydc/bin/centrifydc stop
  /etc/init.d/centrify-sshd stop
  /etc/init.d/adfsagent stop

- Upgrade the core agent packages using the native package manager. The four core packages must be upgraded together.
- Upgrade other Delinea packages using the native package manager.
- Restart Delinea processes or reboot the computer.

Depending on the order in which you are upgrading individual packages, you might see warnings from the package manager about file dependencies. If you see that a dependency is generated because of a package you have yet to upgrade, it is safe to ignore the warning.

The next sections illustrate the commands to use on different platforms. The actual file name that you specify on the command line—including a specific build number, platform, and architecture—will identify the specific operating system you are updating, for example centrifydc-5.4.2-sol8-sparc-local.tgz or centrifydc5.4.2aix53ppc-bff.gz.

Performing Upgrades on UNIX Computers

The process for simultaneous upgrades on UNIX computers is similar to that for Linux computers. However, the native package managers on different platforms vary in their ability to perform simultaneous upgrades.

This section includes the following topics:

Upgrading Packages on Solaris Computers

On Solaris computers, it is necessary to spool all packages that are to be installed simultaneously. The package manager can then take the spooled packages and install them all at once using one command. Before upgrading on Solaris computers, however, you should stop all Delinea processes that are running.

> **Note**: On Solaris 10 computers that use Solaris zones, you should upgrade the core agent packages as a separate step. You can then upgrade other Delinea packages using a simultaneous upgrade.

To perform upgrades on Solaris computers:

1. Stop existing Delinea processes.

   For example, if you are upgrading the core agent, Delinea-enabled OpenSSH, and Delinea NIS packages, you would enter commands similar to the following:

```
/usr/share/centrifydc/bin/centrifydcstop
etc/init.d/centrify-sshd stop
/etc/init.d/adnisd stop (on Solaris 9)
svcadm disable centrifydc_server (on Solaris 10 or later)
```

2. Create a new admin file.

   If you are upgrading an existing installation, make a copy of the system default admin file (/var/sadm/install/admin/default) and modify it to ignore dependencies. In the examples below, this file is called my_admin. It should look like this:

   mail=
   instance=overwrite
   partial=nocheck
   runlevel=nocheck **idepend=nocheck** rdepend=quit
   space=quit
   setuid=nocheck **conflict=nocheck** action=nocheck basedir=default

   If you are performing a fresh installation, you can use the original system admin file and keep the default settings.

3. Unzip and extract each package into a temporary directory, for example, my_tmp_dir.

   To unzip and extract the agent core packages, you would enter commands similar to the following:

   ```
   gunzip centrifydc-5.4.0-*platform-arch*-local.tgz
   tar xvf centrifydc-5.4.0-platform-arch-local.tar

   gunzip centrifydc-openssl-5.4.0-*platform-arch*-local.tgz
   tar xvf centrifydc-openssl-5.4.0-*platform-arch*-local.tar

   gunzip centrifydc-openldap-5.4.0-*platform-arch*-local.tgz
   tar xvf centrifydc-openldap-5.4.0-*platform-arch*-local.tar

   gunzip centrifydc-curl-5.4.0-*platform-arch*-local.tgz
   tar xvf centrifydc-curl-5.4.0-*platform-arch*-local.tar
   ```

4. Spool the packages.

   Spool the packages to a specified directory, for example, my_spool_dir.

   To spool the core packages, you would run commands similar to the following:

   ```
   pkgadd -s /my_spool_dir -d /my_tmp_dir/CentrifyDC CentrifyDC
   pkgadd -s /my_spool_dir -d /my_tmp_dir/CentrifyDC-openssl CentrifyDC-openssl

   pkgadd -s /my_spool_dir -d /my_tmp_dir/CentrifyDC-openldap
   CentrifyDC-openldap
   pkgadd -s /my_spool_dir -d /my_tmp_dir/CentrifyDC-curl CentrifyDC-curl
   ```

5. Upgrade the packages.

   To upgrade the core packages, you would enter commands similar to the following:

   ```
   /usr/sbin/pkgadd -a my_admin -n -d /my_spool_dir CentrifyDC-openssl
   /usr/sbin/pkgadd -a my_admin -n -d /my_spool_dir CentrifyDC-openldap
   CentrifyDC-curl
   /usr/sbin/pkgadd -a my_admin -n -d /my_spool_dir CentrifyDC
   ```

6. Restart Centrify processes after the upgrade is complete.

7. Verify the upgrade.

   To verify that the upgrade was successful, run the following command:

   ```
   /usr/bin/pkginfo \| grep -i Centrify
   ```

Upgrading Packages on HP-UX Computers

On HP-UX computers, it is necessary to spool all packages that are to be installed. The package manager can then take the spooled packages and install them all at once using one command.

To perform upgrades on HP-UX computers

1. Copy and unzip all depot.gz packages into a temporary directory, for example, my_dir.

   To unzip and extract the agent core packages, enter commands similar to the following:

   ```
   gunzip centrifydc-5.4.0-*platform-arch*.depot.gz
   gunzip centrifydc-openssl-5.4.0-*platform-arch*.depot.gz
   gunzip centrifydc-openldap-5.4.0-*platform-arch*.depot.gz
   gunzip centrifydc-curl-5.4.0-*platform-arch*.depot.gz
   ```

2. Spool each package.

   On HP-UX computers, you can use the default spool directory, but you must create a working directory, for example my_dir.

   To spool the agent core packages to my_dir, enter commands similar to the following:

   ```
   swcopy -s /full_path/my_dir/centrifydc-openssl-5.4.0-*platform-arch*.depot
   CentrifyDC-openssl
   swcopy -s /full_path/my_dir/centrifydc-openldap-5.4.0-*platform-arch*.depot
   CentrifyDC-openldap
   swcopy -s /full_path/my_dir/centrifydc-curl-5.4.0-*platform-arch*.depot
   CentrifyDC-curl
   swcopy -s /full_path/my_dir/centrifydc-5.4.0-*platform-arch*.depot
   CentrifyDC
   ```

3. Upgrade the packages.

   Use a single command to upgrade all packages. For example, to update the core agent packages, enter a command similar to the following:

   ```
   swinstall -s CentrifyDC-openssl CentrifyDC-openldap CentrifyDC-curl
   CentrifyDC
   ```

4. Verify the upgrade.

   Verify that the upgrade was successful by running the following commands:

   ```
   swlist \l grep -i Centrify
   swverify CentrifyDC
   ```

Upgrading Packages on AIX Computers

On AIX computers, it is necessary to unzip all packages that are to be installed. The package manager can then take the unzipped packages and install them all at once, using one command.

To perform upgrades on AIX computers do the following:

1. Copy and Unzip the packages to a directory, for example, my_dir.

   If you are upgrading the core agent packages, you would run commands similar to the following:

   ```
   gunzip centrifydc-5.4.0-*platform-arch*-bff.gz
   gunzip centrifydc-openssl-5.4.0-*platform-arch*-bff.gz
   gunzip centrifydc-openldap-5.4.0-*platform-arch*-bff.gz
   gunzip centrifydc-curl-5.4.0-*platform-arch*-bff.gz
   ```

2. Upgrade the packages.

   You can now upgrade the packages using commands similar to the following:

   ```
   inutoc .
   installp -aY -d my_dir all
   ```

Upgrading Agents on Solaris Systems

Before upgrading agents on Solaris systems you'll need the installation packages appropriate for your Solaris system, as listed in the table below.

Solaris Agent Installation Packages

Solaris 10	svr4	x86	centrify-server-suite-2021.1-sol10-x86.tgz
Solaris 10	svr4	Sparc	centrify-server-suite-2021.1-sol10-sparc.tgz
Solaris 11	svr4	x86	centrify-server-suite-2021.1-sol10-x86.tgz
Solaris 11	svr4	Sparc	centrify-server-suite-2021.1-sol10-sparc.tgz
Solaris 11	IPS	x86	centrify-server-suite-2021.1-sol11-i386.tgz
Solaris 11	IPS	Sparc	centrify-server-suite-2021.1-sol11-sparc.tgz

To upgrade the Solaris svr4 packages:

- Follow the instructions here: Using the install.sh shell script to update packages or Performing upgrades on UNIX computers

Upgrading and Migrating Solaris svr4 Packages to IPS on Solaris 11

If you have a Solaris 11 system on which you've installed a previous release of the Delinea Agent for *NIX and you want to upgrade and use Solaris IPS, you do the upgrade in 3 steps:

1. Run the install-last-svr4.sh script.

   This upgrades your previous agent to a temporary, in-between version 9.9.9.

2. Run the install-last-svr4.sh script again and uninstall the version 9.9.9 packages.

   This prepares the system so that you can install the IPS packages.

3. Run the IPS-specific install script, install-ips.sh to install the Solaris IPS packages.

These steps assume that there aren't any child zones configured on this Solaris system.

You'll need the IPS specific tgz file and also the svr4 tgz for your system in order to perform these steps.

Before you upgrade:

- Make sure that the agents components that are installed before you upgrade are the same ones that you want to have installed after the upgrade. It's easier to do the upgrade in place this way.

  For example, if you have the audit agent currently installed but you don't want it installed in the new version, remove it before upgrading. Or, if you don't have the ldap proxy component installed but you do want it installed in the new version, install it first (the same version as the other agent components that are already installed).

To upgrade and migrate the Solaris svr4 packages to IPS on Solaris 11:

1. Log on or switch to the root user.
2. Change to the appropriate directory that contains the Delinea Agent package you want to install.

   For example, change to the Agent_Unix directory.

   If you downloaded individual agent packages from the Delinea Download Center, unzip and extract the contents. For example, you'll need the following two .tgz files:

centrify-infrastructure-services-2021.1-sol11-i386.tgz

centrify-infrastructure-services-2021.1-*SolarisVersion-platform-arch*.tgz

For each package, extract them; here's an example:

gunzip -d centrify-infrastructure-services-*<release>*-platform-arch.tgz

tar -xf centrify-infrastructure-services-*<release>*-platform-arch.tar

3. Change to the /lastsvr4 sub-directory and run the install-last-svr4.sh script to prepare the previous installation for upgrade:
    1. The script prompts you, "Do you want to continue and upgrade to v.9.9.9?"

       Enter Y for yes.
    2. The script prompts you to confirm your selection:

       Enter Y to continue (or enter N to change the settings).

       The script unpackages the files and installs the new packages.
4. Run the install-last-svr4.sh script a 2nd time and uninstall the 9.9.9 version components:
    1. The script prompts you, asking if you want to erase, reinstall, or keep the current package.

       Enter E to erase.
    2. The script prompts you to confirm your selection to uninstall the package.

       Enter Y to continue.

       The script uninstalls and removes the version 9.9.9 files.
    3. You can run the following command to verify the Solaris agent package svr4 installation status:

       pkginfo | grep -i Centrify

       Note that there is no space between "pkg" and "info"; if you search for "pkg info" you'll be searching for IPS packages.
5. Change to the directory (up one level) that contains the installation packages and run the install-ips.sh script.
    1. The script will list that it found previously installed svr4 packages and prompts you, asking if you want to continue.

       Enter Y for yes.
    2. The script will list out the package versions to install and prompts you to confirm.

       Enter Y to confirm and continue.

       The script installs the IPS packages.
6. You can run the following command to verify the Solaris agent package IPS installation status:

   pkg info | grep -i Centrify

   Note: The space between "pkg" and "info"; if you search for "pkginfo" you'll be searching for svr4 packages.

Delinea

Upgrading Managed Mac OS X Computers

In most cases, you can update agents on Mac OS X computers by simply installing the new agent either directly or remotely on top of an existing agent. As a best practice, you should perform in-place upgrades using a local Mac administrative (admin) account or any other user account that has local administrative rights and reboot the computer after completing the upgrade. In most cases, you should not perform the upgrade while you are logged on as an Active Directory user in a currently active session.

In rare cases, you might be advised to run adflush to clear the Active Directory cache before performing an in-place upgrade. For example, if you are updating agents from version 4.x, or earlier, to 5.1.x, run adflush first to ensure a smooth upgrade. It is highly unusual for an upgrade to require you to leave and rejoin a managed Mac computer to the domain.

Compatibility for Additional Packages

In general, Delinea software packages are not version-dependent on each other. However, there are compatibility limitations in some situations. This chapter describes specific compatibility requirements for packages that are not part of the core agent packages or have been added to or removed from the core agent packages. If you are only upgrading the core agent packages and have no other packages installed, you can skip this chapter.

Should You Be Concerned About Compatibility?

Compatibility issues are managed automatically when you use the install.sh shell script to upgrade packages. If you plan to update packages using a native package manager, however, you should be aware of potential compatibility issues and be able to manually manage dependencies between packages. Depending on the version of Delinea software you currently have installed, the version you are upgrading to, and which packages you have installed, you might have many or no compatibility concerns. The first step is to identify which software packages and versions you have deployed.

The core agent package for access control and privilege management for versions before version 5.4.0 is CentrifyDC. For all releases after and including 5.4.0, the core agent package is split into four distinct packages:

CentrifyDC

CentrifyDC-openssl
CentrifyDC-openldap
CentrifyDC-curl

The core agent package for auditing is CentrifyDA. Other packages you might have installed include:

CentrifyDC-nis
CentrifyDC-krb5
CentrifyDC-ldapproxy
CentrifyDC-openssh
CentrifyDC-web
CentrifyDC-apache
CentrifyDC-adbindproxy

> **Note**: When you upgrade to version 5.4.0 of the Delinea Agent, you must also upgrade all of the other Delinea packages you have installed to version 5.4.0 as well. If you fail to upgrade a package other than the core packages, and attempt to upgrade the core agent packages to 5.4.0, the upgrade will fail. For agent versions after 5.4.0, you may not be required to simultaneously upgrade each of the packages other than the core agent packages.

Removing the CentrifyDC-samba Package

If you are upgrading the core agent package for access control and privilege management and have Delinea Samba installed, you should remove the Delinea Samba CentrifyDC-samba) package, install open-source Samba, and install the Delinea adbindproxy package (CentrifyDC-adbindproxy). See the **Samba Integration Guide** for details about that procedure.

Compatibility for CentrifyDC-nis Package

If you are upgrading the core agent packages and have the CentrifyDC-nis package installed, you should also upgrade the CentrifyDC-nis package. The CentrifyDC-nis package must have the same major version number as the core agent packages. The version number for the CentrifyDC-nis package should never be higher than the version number of the core agent packages.

> **Note**: Some platforms, the adnisd package might prevent the ypbind service from starting properly because of the order in which services are started. For example, if ypbind is configured to start before the adnisd service, the bind will fail. This issue does not occur if you are installing new packages. However, to prevent unintended changes to the existing startup sequence during an upgrade, upgrading the adnisd package will not modify your existing startup configuration. You can manually correct the startup sequence after an upgrade by manually running the chkconfig script. For example, run the following command after the adnisd upgrade:

```
chkconfig adnisd on
```

Compatibility for CentrifyDC-krb5 Package

The Delinea Kerberos client tools are no longer packaged with the Server Suite agent or available from the Delinea Download Center. The client tools were formerly provided as a separate software package or as part of the core agent package to support Kerberos-based authentication on older operating systems. The package is no longer relevant on currently-supported operating systems.

Compatibility for CentrifyDC-ldapproxy Package

If you are upgrading the core agent packages and have the CentrifyDCldapproxy package installed, you should also upgrade the CentrifyDCldapproxy package. The CentrifyDCldapproxy package must have the same major version number as the core agent package. The version number for the CentrifyDCldapproxy package should never be higher than the version number of the core agent package. If you upgrade the core agent packages to a version number that is higher than the CentrifyDCldapproxy package version, the installation script removes the CentrifyDCldapproxy package. To retain the CentrifyDCldapproxy package when you upgrade the core agent packages, you must make sure that both packages are upgraded to the same version number.

Compatibility for CentrifyDC-openssh Package

In most cases, the core agent packages and the CentrifyDCopenssh packages are installed and upgraded together. Therefore, in most cases, they will have the same major version number. If you have the CentrifyDCopenssh package installed and are upgrading the core agent to version 5.1.2 or later, you must also upgrade the CentrifyDCopenssh package. If you use the installation script to upgrade, it enforces this compatibility requirement.

Delinea

Compatibility for CentrifyDC-Apache and CentrifyDC-Web Packages

If you are upgrading the core agent packages to 5.x and have Delinea for Apache or Java applications installed, the CentrifyDC-apache or CentrifyDC-web package should be version 4.x or later. For example, CentrifyDC_apache-4.2.0-nnn is compatible with CentrifyDC version 5.x.

Upgrading Version-Dependent Packages

If you are upgrading a computer that has one or more Delinea software packages that are version-dependent on one another, you should either:

- Remove the Delinea packages that are version-dependent before upgrading the core agent packages, upgrade the core agent packages, then re-install the new versions of the version dependent packages.
- Simultaneously upgrade the core agent packages and all of the additional packages that are version-dependent.

If you are upgrading a computer where there are no version dependencies, Delinea recommends you upgrade all packages simultaneously, if possible.

Working with Classic Zones After an Upgrade

Server Suite supports both classic and hierarchical zones. After you upgrade the agents, you can choose to either migrate your classic zones into a hierarchical zone structure or maintain them as classic zones. If you want to convert your classic zones into hierarchical zones, you can use the admigrate program. For details about using the admigrate program to migrate a classic zone to a new parent or child hierarchical zone, see the man page for admigrate.

> **Note**: You can only migrate classic zones to hierarchical zones if you have upgraded the Delinea Agent to version 5.x or later.

You are not required to migrate any existing classic zones. If you choose to maintain your existing zones as classic zones, however, you should be aware that the authorization model in classic zones differs from the authorization model used in hierarchical zones. For example:

- In classic zones, any user with a profile in a zone is automatically granted login access to all computers joined to the zone.
- In hierarchical zones, a user with a profile in a zone must be assigned to a role with login rights and PAM access rights before being able to login to a computer joined to a zone.

In addition, there are configuration parameters, commands, APIs, and features that are only applicable in classic zones and other parameters, commands, APIs, and features that are only applicable in hierarchical zones. For example, authorization is an optional feature that can be enabled or disabled in classic zones, so there is a configuration parameter and a zone property option to support the feature in classic zones. For hierarchical zones, authorization is required for access to any managed computer, so the configuration parameter and zone property option are not visible in hierarchical zones.

What To Do If There Are Problems During an Upgrade

In most cases, upgrading Delinea software is a seamless process that does not interrupt services. If you are not able to complete an upgrade successfully, however, there are a few steps you can take to restore your working environment. This chapter covers the steps to take if you have problems during the upgrade process.

Delinea

Remove and Re-install Authentication and Privilege

If you have problems upgrading any Authentication & Privilege components, such as Access Manager, you should use the Control Panel application to uninstall the software, then rerun the setup program to install the components cleanly.

If you want to restore an older version of the software rather than attempt a fresh installation of the latest version—run the setup program for that version of the software.

Remove and Re-install Delinea Audit and Monitoring Service

If you have problems upgrading any Delinea Audit and Monitoring Service components, such as Audit Manager or Audit Analyzer, you should do the following:

- Use the Control Panel application to remove the auditing infrastructure components from the local computer.
- Use ADSI Edit to remove the service connection point for the installation. If you publish this information in more than one location, remove all of the service connection points from the forest.
- Rerun the setup program to install the components cleanly.

If you want to restore an older version of the software rather than attempt a fresh installation of the latest version—run the setup program for that version of the software.

Remove and Re-install Agent Features

If you have problems upgrading any agent features, such as access control and privilege management or auditing services, you should do the following:

- Log on as root and disable auditing on UNIX computers where auditing is enabled:
    1. dacontrol -d
- Use the adleave command to remove the UNIX computer from its current zone and Active Directory domain.
- Use the Delinea Privilege Elevation Service settings to remove the local Windows computer from its current zone, then use the Windows Control Panel application to remove the agent services from the local computer.
- Rerun the install.sh script or the agent setup program to install the agent cleanly.
    1. You can join the domain from the installation script on UNIX computers or join a zone from the agent configuration wizard on Windows computers.
- Log on as root and enable auditing on UNIX computers where you want auditing enabled.
    1. dacontrol -e

Delinea

Known Issues

The following are known issues organized by category.

Installation and Uninstallation Issues

- Upgrading from the beta build to this version may result in offline MFA mode if there are multiple authentication servers registered in your AD forest. To resolve this, uninstall the beta build first and then install this new version. (Ref: CS-41915)
- The Delinea Common Component should be the last Server Suite component uninstalled. If the component is uninstalled before other component, it must be reinstalled by the uninstall process to complete its task. (Ref: 36226a)
- If you intend to install the software on the desktop with elevated privilege, you should not check the "Run with UAC restrictions" option when creating the desktop. (Ref: 39725b)
- When you double-click on the Delinea Agent for Windows msi and select the "repair" option, the existing files are replaced irrespective of their version number, even when they are identical. As a result, a prompt to restart the system is displayed as files that were in use were replaced. However, if you use the Easy Installer to do the repair and a file on the disk has the same version as the file that is part of the installer package, the installed file will not be replaced. Therefore, there will not be any prompt to restart the system. (Ref: 26561a)
- If you uninstall the Delinea Agent for Windows while the DirectAudit Agent Control Panel is open, files needed by the uninstall process may be blocked. You should close the DirectAudit Agent Control Panel for a successful conclusion to the uninstall process. (Ref: 25753a)
- Agent for Windows and its installer are built on .NET. Therefore, .NET is always installed as a pre-requisite before the agent is installed. If .NET is removed from the system later, Delinea Agent for Windows will not run properly. User will also experience problem when trying to remove Delinea Agent for Windows from the system. To properly uninstall Delinea Agent for Windows, please make sure Delinea Agent for Windows is uninstalled before .NET. (Ref: 39051a)

Configuration Issues

- In a cross-forest environment, forest A user cannot enroll a device joined to forest B when forest A does not have a connector. (Ref: CS-44805)
- In Windows 2016 and Windows 10, during the login process, selecting SMS or using other mechanisms like Security Question/Phone call/Password/Email/Mobile for MFA and clicking the **Commit** button will be intermittently unresponsive. (Ref: CS-41699)
- In some large environment with multiple domain controllers, it may take up to one minute for the new zone setting in Delinea Agent Configuration to take effect. (Ref: 58128b)
- If one of the Global Catalog servers is unavailable, user may not be able to configure the zone for Delinea Agent for Windows. (Ref: 58621b)
- Microsoft normally automatically distributes and installs root certificates to the Windows system from trusted Certificate Authorities (CA) and users are seamlessly able to use a secure connection by trusting a certificate chain issued from the trusted CA. However, this mechanism may fail if the system is in a disconnected environment where access to Windows Update is blocked or this feature of automatic root certificate installation is disabled. Without updates on the certificate trust list (CTL), the default CTLs on the system may not be adequate for secure connections of multi-factor authentication especially for older versions of Windows such as Windows 7 and Windows Server 2008 R2. To ensure the success of multi-factor authentication, user may need manually distribute and install the latest CTLs and the required root certificate to systems in a disconnected environment. See Delinea KB-6724 for further information. (Ref: CS-39703)

Environment Issues

- On Windows 10 and Windows 2016 machines with Delinea Privilege Elevation Service, the following will occur (Ref: CS-43883):
  - Pop up an error dialog several seconds after clicking "Open file location" in the context menu of a shortcut on the start menu. Explorer windows will display correctly.
  - No responses to the following actions * Clicking "Open file location" in the context menu of a shortcut on desktop * Clicking "Open file location" in the context menu of a shortcut on the Delinea Start menu in the Privileged Desktop * Slow response to "OK", "Cancel" in the shortcut property page after "Open file location" in the general tab is clicked. The dialog will close after several seconds.
- On some Windows 10 computers, the smart card login option may not be displayed if another credential method has been recently used. To display the smart card login option, remove and insert a smart card into the reader. This will cause the login screen to reload and will display the smart card login option. (Ref: CS-41282)
- An environment with no Global Catalog is not supported. (Ref: 46577a)
- Delinea Privilege Elevation Service requires machine time to be synchronized with domain controller. VMware virtual machine has a known issue that its time may not be synchronized with domain controller. This problem occurs more often on an overloaded virtual machine host. If the system clocks on the local Windows computer and the domain controller are not synchronized, Delinea Privilege Elevation Service does not allow any domain users to login. You can try the following KB from VMware to fix the time synchronization issue. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1189 (Ref: 47795b)

RunAsRole Issues

- If you use the "RunAsRole.exe /wait" command to run a Python script, the input/output cannot be redirected for versions of Python below 3.0.0. (Ref: 45061a)
- The Run As Role menu is not available on the start screen in Windows 8 or Windows 2012 or later because Microsoft doesn't support any custom context menu on the start screen. User has to go to the Windows desktop in order to launch an application using Run As Role context menu. (Ref: 35487a)
- When running "RunAsRole.exe /wait sc.exe" with no argument provided to sc.exe, sc.exe will prompt
  - Would you like to see help for the QUERY and QUERYEX commands? [y | n]:
  - Typing 'y' or 'n' doesn't do anything because the input cannot be successfully redirected to sc.exe. (Ref: 47016b)
- It is not recommended to change zone via Run As Role since the role that is in use may no longer be available once after leaving from the previous zone during the change zone process. (Ref: 58043a)
- On Windows Server 2008 R2 and Windows 7, if the Agent machine has no internet connection and the .NET CLR settings (checkCertificateRevocationList) is set to True, the MFA authentication will be failed because the CLR is unable to verify the certificate through internet. The workaround is to enable the internet connection or turn off the CLR settings (set checkCertificateRevocationList to False which is also the default value). (Ref: CS-40147)

Desktop with Elevated Privileges Issues

- On a desktop with elevated privileges, if you use “Control Panel > Programs > Programs and Features” to uninstall a program, you may see the following warning message and cannot uninstall the software.

  "The system administrator has set policies to prevent this installation."

  This issue happens when User Account Control (UAC) is enabled and when “Run with UAC restrictions” is selected when creating the new desktop. (Ref: 33384a)

- You cannot use the Start menu option “Switch User” while you are using a role-based, privileged desktop. To use the “Switch User” shortcut, change from the privileged desktop to your default Windows desktop. From the default desktop, you can then select Start > Switch User to log on as a different user. (Ref: 39011b)

Roles and Rights Issues

- There is no 'Require multi-factor authentication' system right for the predefined 'Windows Login' role. To define this system right for MFA, use the pre-defined Require MFA for logon role, or create a new custom role. (Ref: CS-40888)
- Windows Network Access rights do not take effect on a Linux or UNIX machines. If you select a role to start a program or create a desktop that contains a Network Access right, you can only use that role to access Windows computers. The Windows computers you access over the network must be joined to a zone that honors the selected role. The selected role cannot be used to access any Linux or UNIX server computers on the network. (Ref: 32980a)
- Network Access rights are not supported on the Windows 2008 R2 Terminal Server if "RDC Client Single Sign-On for Remote Desktop Services" is enabled on the client side. (Ref: 34368b)
- To elevate privileges to the "Run as" account specified in a Windows right, the "run as" account must have local logon rights. If you have explicitly disallowed this right, you may receive an error such as "the user has not been granted the requested logon type at this computer" when attempting to use the right. (Ref: 34266a)
- If your computer network is spread out geographically, there may be failures in NETBIOS name translation. If a NETBIOS name is used, Active Directory attempts to resolve the NETBIOS name based on the domain controller that the user belongs to, which in a multi-segment network might fail. Therefore, Network Access rights might not work as expected if the remote server is located using NETBIOS name. You may need to consult your network administrator to work around this issue. (Ref: 39087a)
- File hash matching criteria in the Application right is not supported for a file larger than 500MB. This is to make sure DirectAuthorize does not spend too much CPU and memory resources to calculate the file hash. User trying to import a file with the size larger than 500MB will see an empty value for the file hash field. (Ref: 56778a)
- For a small set of application, enabled matching criterion - "Product Name", "Product version", "Company", "File Version" or "File Description" of a Windows Application Right may fail to match after upgrading agent under the following conditions: - Any value for the enabled matching criteria is defined by either import from a process or file - The matching criteria is defined by 5.1.3 or 5.2.0 DirectManage Access Manager since the number of affected application is expected to be relatively low, proactively updating the defined matching criteria of Windows Application Right is not necessary. (Ref: 60053a)

Compatibility with Third Party Products Issues

- VirtualDesktop is not compatible with Delinea Agent for Windows. Users should use the Delinea system tray applet to create virtual desktop instead. (Ref: 44641b)
- The startup path for "SharePoint 2010 Management Shell" and "Exchange Management Shell" may set to C:\Windows instead of user home directory if it is launched via RunAsRole.exe or from a desktop with elevated privilege. (Ref: 38814b, 46943b)
- Attempting to enable Kerberos authentication for Oracle databases will fail. This issue is being brought to the attention of Oracle Support for a resolution in upcoming releases. (Ref: 33835b)
- Some applications do not use the process token to check the group membership. They check the user's group membership on its own. Therefore, any Windows rights configured to use a privileged group will not take effect in these applications. The workaround is to use a privileged user account instead of a privileged group. Here is the list of known application with this issue:
    - vCenter Server 5.1
    - SQL Server
    - xchange 2010 or above
    - SCOM 2007

      (Ref: 45318a, 45218a, 43779a, 38016a)
- Users may notice an error and cannot install ActivClient after installing Delinea Agent for Windows. During the installation of ActivClient, it attempts to change the local security setting. However, there is a known issue for Delinea Agent for Windows of blocking the local security setting (Ref: 63609b). Therefore, users may not be able to install ActivClient successfully after installing Delinea Agent for Windows. We suggest installing ActivClient before installing Delinea Agent for Windows. If Delinea Agent for Windows has been installed, please uninstall it and follow the installation sequence suggested. This issue happens on Windows 8.1 and Windows 2012 R2 only. (Ref: 76016b)

Application Manager Issues

Application Manager does not support the Server Core edition of Windows. (Ref: CS-40656).

Configuration

- Unix Configuration Guide
- Group Policy Admin Guide
- NIS Admin Guide
- Smart Card Configuration Guide

The *Configuration and Tuning Reference Guide* provides reference information for Server Suite configuration parameters. You can set configuration parameters locally on Linux, UNIX, and Mac OS X computers to fine tune the operation of Server Suite components and subsystems. Server Suite is an integrated software solution that delivers secure access control and centralized identity management through Microsoft Active Directory. With Server Suite software, your organization can improve IT efficiency, regulatory compliance, and security for on-premise, mobile, and hosted resources.

Intended Audience

This guide is intended for administrators who want to customize the operation of Server Suite components and subsystems by modifying locally-defined configuration parameters. Many of these operations can also be configured remotely using group policies. This guide is intended as a supplement to the main Server Suite documentation set. It assumes that you have a working knowledge of Server Suite components and administration.

For information about planning a deployment and installing components, see the *Planning and Deployment Guide*. For information about performing administrative tasks using Access Manager, see the *Administrator's Guide for Linux and UNIX*.

Limitations of this Guide

This guide is updated with every major release of Server Suite. Because the supported configuration parameters can change from one release to another, have different default values between releases, or be designed to address very specific conditions, you should consider the configuration files (centrifydc.conf and centrifyda.conf for example) included with the software to be the definitive source of information for the parameters in the version of the software you are using. If there are differences between the information in the configuration files and this guide, you should consider the comments in the configuration file itself to be the most current or accurate for your environment.

The sections in this guide are as follows:

Working with Parameters and Agent Configuration Files

The Delinea Agent configuration files, centrifydc.conf and centrifyda.conf, can be used to customize and control the operation of Server Suite components and subsystems on a local host computer. This section provides an introduction to using the configuration file and setting values for the configuration parameters defined in the file.

Controlling agent operations

The Delinea configuration file for access control and privilege management is /etc/centrifydc/centrifydc.conf. The Delinea configuration file for auditing is /etc/centrifyda/centrifyda.conf. Depending on the deployment options selected when you install the agent, one or both of these files might be available on each Delinea-managed computer. The configuration files contain parameters that specify how Delinea components and subsystems operate on the local computer. They can be used to tune operations to suit your environment, for example to address bandwidth or latency constraints or address specific requirements, for example, to prevent the storage of a password hash. Many of the operations controlled locally by configuration parameters can also be controlled remotely using group policies. For information about customizing operations using group policies, see the *Group Policies Guide*.

You only have to edit the /etc/centrifydc/centrifydc.conf or /etc/centrifyda/centrifyda.conf file if you want to set custom values for one or more configuration parameters. For most organizations, the default values are appropriate. However, if you decide that you want to use a custom value for any parameter, you can uncomment the parameter name in the appropriate configuration file, then set an appropriate parameter value in place of the default value.

> **Note**: In most cases, you only modify settings in the configuration files if you want to customize specific behavior locally on an individual computer. For most parameters, you can make changes, then run the adreload command to have the changes take effect immediately. Some parameters, however, require you to restart the agent (adclient). Similarly, if you make changes to the configuration parameters used by the Delinea Network Information Service (adnisd), you may need to run the adreload command or restart that service.

Basic syntax used in configuration files

Configuration parameter are defined using a key/value pair that identifies the configuration parameter name and the value assigned to that parameter. If a configuration parameter is not explicitly set in the configuration file, the Delinea Agent assumes a default value for that parameter.

A key/value pair in the configuration file typically takes the following form:

parameter_name: value

where parameter_name is the name of the configuration parameter that describes the component the setting applies to or the purpose of the parameter, and value is the value assigned to that parameter. Variations in the formatting of the key/value pair are allowed. For example, the parameter name can be followed by a colon (:), equal sign (=), or a space:

parameter_name=value

parameter_name value

Setting configuration parameter names

In most cases, parameter names are fixed strings that are defined in the default centrifydc.conf file and commented out to illustrate the default value or how to configure a setting. In some cases, however, the parameter name itself must be customized to enable a setting. For example, the configuration parameter pam.mapuser.localuser must include the specific local user name you are mapping to an Active Directory account. For example, to map the local user joan7 to the Active Directory user joan.adams, you must set the parameter name to pam.mapuser.joan7 to specify that the mapping is for the local user joan7:

pam.mapuser.joan7: joan.adams

Setting configuration parameter values

Depending on the configuration parameter you are setting, the parameter value can be a string, a numeric value, or a Boolean value. For example, user names and group names defined in Active Directory are specified as strings using a valid Active Directory form, such as user[@domain]. In some cases, string parameter values can include environment variables.

In general, you can specify user names in the configuration file with any of the following valid formats:

- Standard Windows format: domain\user_name
- Universal Principal Name (UPN): user_name@domain
- Alternate UPN: alt_user_name@alt_domain
- UNIX user name: user

However, you must include the domain name in the format if the user account is not in the local computer's current Active Directory domain. In addition, if you are specifying an Active Directory logon name that contains spaces, you should use quotes around the string. For example:

adclient.hash.allow: 'marco sanchez@arcade.com'

Using special characters

Configuration parameter values can include the following special characters that are often used in UNIX scripts:

- The dollar sign ($) signifies an environment variable that can be resolved to an appropriate value if recognized by the agent. Valid environment variable names can consist of alphanumeric characters and underscores.
- A backslash (\) signifies that the next character is a literal, and is used, among other things, to specify a trailing space (\ ) or a single backslash (\\).

Boolean values are case-insensitive. The permissible values are true, yes, false, and no.

If a parameter can take multiple values, those values are separated from each other by a comma or a space. Spaces preceding or trailing each value are ignored.

Using environment variables

The values in key/value pairs can include standard shell environment variables. The variables are resolved to their current value when the Centrify Agent reads the configuration file. For example, you can use the environment variable $HOSTNAME to include the local computer's host name in any parameter value setting:

example_parameter: test_$HOSTNAME

If the name of the current managed computer is host1, the configuration parameter example_parameter takes the value test_host1.

In addition to standard environment variables, you can use the following Centrify-specific environment variables in the configuration file:

- $ZONE is the name of the host computer's Centrify zone.
- $JOINNAME is the name of the host computer's account name in Active Directory.
- $DOMAIN is the name of the Active Directory domain to which the host computer is joined.
- $SITE is the name of the Active Directory site for the host computer.

Rereading parameter values after making changes

In most cases, you can either run the adreload command or restart the agent (adclient) to have changes to any configuration parameter take effect. Running the adreload command or restarting the adclient process forces the Delinea Agent to reread the configuration parameters that have been defined, including any values that have changed since the last time the configuration file was read.

For most configuration parameters, you can run the adreload command to have changes take effect without restarting the adclient process. There are a few configuration parameters, however, that cannot be reloaded by running the adreload command. If you want to ensure that the agent rereads all configuration parameters, you should restart the adclient process. For example, to ensure all changes to adclient-related configuration parameters take effect, you can restart the adclient process.

Similarly, if you make changes to the configuration parameters used by the Delinea Network Information Service (adnisd), you can either run the adreload command or restart the adnisd service to ensure those changes take effect. If you change LRPC- or NSS-related parameters, you should restart both the adclient and adnisd processes if both are running when you make the change.

Securing parameter settings

By default, the configuration files—centrifydc.conf and centrifyda.conf—are owned by root. In most cases, therefore, the parameter settings you specify are secure because they can only be set or modified by the root user and access to the root account is tightly controlled. However, there are many parameters that allow you to specify settings in an external file. For example, the pam.allow.groups parameter allows you to specify a list of groups in an external file, then set the parameter value to use the file: keyword and the file path and file name of that external file.

If you are using an external file to configure parameter settings, you should ensure that the external file meets the following security requirements:

- The external file is owned by root or an equivalently-protected account.
- The external file is not group or world writable.
- The path you specify to the external file is not a symbolic link.

Using group policies to configure settings

Many configuration parameter values can be controlled by enabling and applying corresponding Delinea group policies through the Group Policy Management Editor. When you use group policies to set configuration parameters, the group policy setting overrides any local configuration setting and the group policy setting is reapplied if the computer is rebooted and periodically when the group policy is automatically refreshed. Therefore, if a group policy exists for configuring a specific setting, in most cases, you should use the group policy rather than edit the local configuration file.

If no group policy exists for a configuration parameter you want to change or if no group policy is applied to the local computer, you can customize the local configuration file to set configuration parameter values as needed.

To determine whether a group policy exists to configure a specific setting, and which group policies affect which settings, see the *Group Policy Guide*. When you open the Group Policy Guide PDF file, use the PDF reader search function to search for a setting (for example, adclient.cache.expires.gc). If the setting can be configured with a group policy, the setting is referred to in the group policy description.

Note: It is possible for an Active Directory administrator to override virtually any setting in the local configuration file using group policies applied to a local computer. This effectively gives administrators with permission to enable or disable group policies root-level access to computers in the zones they manage. There is no way to effectively prevent settings from being changed, except by disabling user, computer, or all group policies in the local centrifydc.conf or centrifyda.conf file or by strictly controlling who has permission to enable and apply group policies to computers that join an Active Directory domain.

For information about disabling the application of group policies using settings in the local centrifydc.conf file, see Customizing group policy configuration for more information about enabling and applying group policies rather than setting configuration parameters locally on a computer, see the *Group Policy Guide*.

Parameters and values are subject to change

Configuration parameters are added, updated, and retired with each release of the Delinea Agent. In addition, some parameters are intended only for specific circumstances and are intentionally not documented in this guide. If a configuration parameter setting is recommended by Delinea Support, but not documented in this guide, you should consider the recommendation made by Support to be authoritative. You should also consider the comments in the configuration file to be the most authoritative reference for the release of the software you are using. Because parameters are often created to address specific issues in specific environments, the default values and recommendations for changes to the default values are also subject to change from one release to another.

Customizing adclient Configuration Parameters

This section describes the configuration parameters that affect the operation of the core agent (adclient) process on the local host computer.

adclient.altupns

This configuration parameter specifies a UPN suffix that adclient will recognize as a valid UPN suffix even if it is a realm unknown by Kerberos.

The default value is “mil”.

For example, to specify “biz” as a suffix to recognize:

adclient.altupns: biz

You can also use multiple UPN suffixes separated by a space. For example, to specify “biz” and “mil” as suffixes to recognize:

adclient.altupns: biz mil

This parameter does not support wildcards (*.acme.com) or preceding dots (.acme.com).

adclient.autoedit

This configuration parameter specifies whether the agent is allowed to automatically edit the NSS and PAM configuration files on the local computer.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value is set to true to allow the files to be edited or false to prevent the files from being edited. The following example allows both the NSS and PAM configuration files to be edited automatically:

adclient.autoedit: true

In most cases, this parameter should be set to true to allow the agent to maintain configuration files automatically. When this parameter is set to true, you can further control the specific individual files to be automatically edited in different operating environments through additional configuration parameters. For example, you can use the adclient.autoedit.nss to enable or disable automatic editing of the nsswitch.conf file or the adclient.autoedit.pam to enable or disable automatic editing of the PAM configuration file. These additional configuration parameters are ignored if the adclient.autoedit parameter is set to false. For more information about the configuration parameters to control the editing of specific files on different platforms, see Enabling automatic editing for specific files.

If you set the adclient.autoedit parameter to false, you must manually edit the appropriate configuration files to enable agent operation. For example, if you set this parameter to false, you should manually edit the nsswitch.conf and /etc/pam.d/systemauth or /etc/pam.d files to include Delinea information or authentication through Active Directory will fail and you may disable login access entirely.

If you want to manually edit the configuration files, you should first make a backup copy of the existing files. After you make a backup copy of the files, you can use the following examples to manually update the files with the configuration information for the agent.

> **Note**: If the adclient.autoedit parameter is not defined in the configuration file, its default value is true.

Enabling automatic editing for specific files

If you set the adclient.autoedit parameter to true, you can use the following parameters to identify the specific files to be automatically edited in different operating environments:

adclient.autoedit.nss	Specify whether you want to automatically edit the Name Service Switch configuration (nsswitch.conf) file on **HP-UX**, **Solaris**, and **Linux** computers. For example: adclient.autoedit.nss: true You can also use group policy to set this parameter.
adclient.autoedit.pam	Specify whether you want to automatically edit the PAM configuration (pam.conf file or pam.d directory) on **AIX**, **HP-UX**, **Solaris**, **Mac OS X**, and **Linux** computers. For example: adclient.autoedit.pam: true You can also use group policy to set this parameter.
adclient.autoedit.centrifypam	Specify whether to activate the Delinea authorization plug-in and add it to the authorization mechanism every time adclient starts. The default value is true. For example: adclient.autoedit.centrifypam: true
adclient.autoedit.centrifypam.restart.securityagent	Specify whether to restart SecurityAgent after the authorization database is edited. The default value is true. For example: adclient.autoedit.centrifypam.restart.securityagent: true If this parameter is set to false, you must restart the SecurityAgent process or reboot the computer manually after the authorization database is edited. If you do not restart SecurityAgent or reboot, users might not be able to log in.
adclient.autoedit.nscd	Specify whether you want to disable automatic editing of the nscd passwd and group cache (nscd.conf) on **Solaris** and **Linux** computers. By default, this parameter is set to false, which means automatic editing is disabled. Setting this parameter to true enables automatic editing. It is recommended that you change the default setting and enable automatic editing of the nscd caches; doing so reduces NSS response time substantially for large volumes of repeated queries.

Note: Some operating systems do not install nscd by default; be sure that nscd is installed before configuring this setting. For example: adclient.autoedit.nscd: false You can also use group policy to set this parameter. | | adclient.autoedit.methods | Specify whether you want to automatically edit the Loadable Authentication Module (LAM) methods.cfg configuration file on **AIX** computers. For example: adclient.autoedit.methods: true You can also use group policy to set this parameter. | | adclient.autoedit.user | Specify whether you want to automatically edit the /etc/security/user file. The default value is true. For example: adclient.autoedit.user: true You can also use group policy to set this parameter. | | adclient.autoedit.user. root | Specify whether root login is controlled by the Centrify authentication mechanism. If this parameter is set to true, the root stanza 'SYSTEM = "compat"' in /etc/security/user will be commented out and root login must go through theCentrify authentication mechanism. The default value is false (so that by default, root login does not go through the <MadCap:variable name="server-company-vars.company-short-name" /> authentication mechanism). For example: adclient.autoedit.user.root: false | | adclient.autoedit.pwgrd | Specify whether you want to automatically edit the password and group hashing and caching daemon (pwgrd) on **HP-UX** computers. For example: adclient.autoedit.pwgrd: true You can also use group policy to set this parameter. |

Note that if you make any changes to any adclient.autoedit.* parameter, you must restart the adclient process for the change to take effect. Restarting adclient is required whether you set the parameters manually in the configuration file or by enabling a group policy.

Related topics

Editing the NSS configuration manually

Editing the PAM configuration manually

Editing the NSS configuration manually

To manually edit the NSS configuration, modify the /etc/nsswitch.conf file to include centrifydc as the first entry for the password and group lines as appropriate for your environment. For example:

passwd: centrifydcfiles
shadow: centrifydcfiles
group: centrifydcfiles

By placing centrifydc at the beginning of each line, you ensure that Active Directory authentication takes precedence over other forms of authentication.

Editing the PAM configuration manually

In most cases, you should not manually edit the PAM configuration on a computer unless absolutely necessary because changes can produce unexpected and undesirable results. If you choose to edit the file manually, you should use caution and limit the changes you make.

To manually edit the PAM configuration to use Delinea and Active Directory, you need to add several lines to the top of the appropriate PAM configuration file for the local operating environment.

For example, on Linux you need to add the following lines to the top of the /etc/pam.d/system-auth file:

auth sufficient pam_centrifydc.so debug
auth requisite pam_centrifydc.so deny debug
account sufficient pam_centrifydc.so debug
session sufficient pam_centrifydc.so homedir
password sufficient pam_centrifydc.so try_first_pass
password requisite pam_centrifydc.so deny

On Solaris and other platforms, you need to add the following lines to the top of the /etc/pam.conf file:

rlogin auth sufficient pam_centrifydc.so debug
rlogin auth requisite pam_centrifydc.so deny debug
login auth sufficient pam_centrifydc.so debug
login auth requisite pam_centrifydc.so deny debug
passwd auth sufficient pam_centrifydc.so try_first_pass debug
passwd auth requisite pam_centrifydc.so deny debug
other auth sufficient pam_centrifydc.so debug
other auth requisite pam_centrifydc.so deny debug
cron account sufficient pam_centrifydc.so debug
other account sufficient pam_centrifydc.so debug

other password sufficient pam_centrifydc.so debug
other session sufficient pam_centrifydc.so debug

> **Note**: In most operating environments, when new users log on successfully, the Delinea Agent automatically attempts to create the user's home directory. In Solaris environments, however, the home directory is often automounted over NFS, so the attempt to automatically create a new home directory for new users typically fails. If you use NFS to automount home directories, you can turn off the automatic creation of the home directory by setting the pam.homedir.create parameter in the centrifydc.conf file to false. For more information about setting this parameter, see pam.homedir.create.

By adding the appropriate lines to the beginning of the PAM configuration file, you ensure that Active Directory authentication takes precedence over other forms of authentication.

Editing the LAM configuration manually

To manually edit the LAM configuration for AIX computers, you need to add Delinea specific information to the /usr/lib/methods.cfg and /etc/security/user files.

In the /usr/lib/methods.cfg file, add the following lines to enable authentication through the Delinea Agent and Active Directory:

CENTRIFYDC:
program = /usr/lib/security/CENTRIFYDC program_64 = /usr/lib/security/CENTRIFYDC64
options = noprompt

In the /etc/security/user file, you need to change the SYSTEM attribute for your users. The easiest way to do this is to change the SYSTEM attribute in the "default" stanza. For example:

...
SYSTEM = "CENTRIFYDC OR CENTRIFYDC[NOTFOUND] AND compat"
...

In addition, if any user has an explicit setting for the SYSTEM attribute, you should remove the setting. For example, by default, the root account has an explicit SYSTEM setting, so you should delete this line or comment it out.

adclient.binding.dc.failover.delay

This configuration parameter specifies the time, in minutes, to wait before adclient fail over to the next domain controller when the currently-connected domain controller is either down or not responding. The default is fail over immediately.

> **Note**: This configuration parameter only takes effect when adclient is running. If the domain controller stored in kset is down when adclient is not running, starting up adclient will force adclient to lookup a healthy domain controller.

adclient.binding.idle.time

This configuration parameter specifies the maximum number of minutes to allow as idle time when binding to Active Directory.

For example, to allow a maximum idle time of 5 minutes during a bind operation:

adclient.binding.idle.time: 5

adclient.binding.ldapsearch.statistic.interval

Use this parameter to specify how much time passes before adclient resets the LDAP search statistics.

The default is 30 minutes.

You can see these statistics by running the following command or by viewing the messages sent to the Unix syslog (centrifydc.log):

adinfo --sysinfo adagent

The adclient.heartbeat.interval parameter controls the heartbeat log messages.

adclient.binding.refresh.force

This configuration parameter specifies whether to force LDAP bindings to be refreshed even if the current binding is to a local (preferred) Active Directory site. Under some conditions, binding to a different site can help facilitate load balancing between servers.

If you set this parameter to true, the agent will attempt to connect to another local domain controller when the period specified in adclient.binding.refresh.interval expires.

By default, this configuration parameter is set to false. For example:

adclient.binding.refresh.force: false

adclient.binding.refresh.interval

This configuration parameter specifies how often to refresh the LDAP bindings to the preferred Active Directory site under these conditions:

- If the computer is currently bound to a local domain controller, bindings are refreshed only if adclient.binding.refresh.force is set to true.
- If the computer is currently bound to a domain controller in another site, bindings are refreshed regardless the adclient.binding.refresh.force setting.

If the agent is unable to communicate with a local domain controller, it automatically connects to an available domain controller in another site until a domain controller in its preferred site becomes available. To determine when a domain controller in the preferred site is available, the agent periodically attempts to re-connect to domain controllers in its preferred site whenever it is connected to a backup domain controller in another site. This parameter controls how frequently the agent performs the attempt to re-connect to the preferred site.

The parameter value specifies the number of minutes between refresh attempts. It must be an integer greater than zero. The following example sets the interval time to 60 minutes:

adclient.binding.refresh.interval: 60

If this parameter is not defined in the configuration file, its default value is 30 minutes.

In changing this parameter, you should consider your network and site topology and the reliability of your servers. If you have highly reliable network links and very good connections between sites, you may find it safe to increase this value, but if communication between sites is slow you should keep this interval short to ensure the agent communicates with domain controllers in its preferred site as soon as possible.

adclient.get.builtin.membership

This configuration parameter determines whether the agent checks for valid users in built-in Active Directory groups, such as Administrators. By default, this parameter's value is false, in which case, the adclient process ignores members of built-in groups.

To include members of built-in groups, set this parameter to true in the configuration file:

adclient.get.builtin.membership: true

adclient.cache.cleanup.interval

This configuration parameter specifies how often the agent should clean up the local cache. At each cleanup interval, the agent checks the cache for objects to be removed or expired, and at every 10th interval, the agent rebuilds local indexes. This parameter's value should be less than the values specified for the adclient.cache.negative.lifetime, adclient.cache.flush.interval, and adclient.cache.object.lifetime parameters.

The default cleanup interval is 10 minutes.

For example:

adclient.cache.cleanup.interval: 10

adclient.cache.encrypt

This configuration parameter specifies whether you want to encrypt the local cache of Active Directory data. If you set this parameter to true, all of the Active Directory data stored in the cache is encrypted and the cache is flushed each time the agent starts up. If you set this parameter to false, the cache is not encrypted and is not flushed when the agent starts up.

For example, to encrypt all data in the cache:

adclient.cache.encrypt: true

If this parameter is not defined in the configuration file, its default value is false.

adclient.cache.encryption.type

This configuration parameter specifies the type of encryption to use when encrypting the local cache. The encryption type you specify must be a type supported in the Kerberos environment. For example, Windows Server 2003 Kerberos supports the following cryptographic algorithms: RC4-HMAC, DES-CBC-CRC and DES-CBC-MD5.

For example:

adclient.cache.encryption.type: des-cbc-md5

This configuration parameter is only used if adclient.cache.encrypt is set to true. If the adclient.cache.encrypt parameter is set to false, this configuration parameter is ignored.

adclient.cache.expires

This configuration parameter specifies the number of seconds before an object in the domain controller cache expires. This parameter controls how frequently the agent checks Active Directory to see if an object in the cache has been updated.

Every object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you aren't using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time to 600 seconds (10 minutes):

adclient.cache.expires: 600

If this parameter is not defined in the configuration file, its default value is 3600 seconds (1 hour).

> **Note**: The adclient.cache.expires parameter defines the default cache expiration time for all objects types. You can override this default value for specific object types by appending the object type to the parameter name. For example, if you want to explicitly override the default expiration time for computer objects, you can define a different value for the adclient.cache.expires.computer parameter. The valid object types you can append to the parameter name to override the default value are: computer, extension, gc, group, search, user, user.membership and zone. Note that adclient.cache.expires.gc, if not set, does not default to the value of adclient.cache.expires, but has its own default value.

adclient.cache.expires.computer

This configuration parameter specifies the number of seconds before a computer object in the domain controller cache expires. If this parameter is not specified, the generic object cache expiration value is used.

Every computer object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for computer objects to 600 seconds (10 minutes):

adclient.cache.expires.computer: 600

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.computer parameter, its value overrides the default value for cached objects.

adclient.cache.expires.extension

This configuration parameter specifies the number of seconds before an extension object in the domain controller cache expires. If this parameter is not specified, the generic object cache expiration value is used.

Every object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for extension objects to 1800 seconds (30 minutes):

adclient.cache.expires.extension: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.extension parameter, its value overrides the default value for cached objects.

adclient.cache.expires.gc

This configuration parameter specifies the number of seconds before information in the global catalog cache expires. The global catalog cache contains the distinguished name (DN) for each object that has been looked up in Active Directory. The primary purpose of the global catalog cache is to store the results from paged object searches. Object attributes are stored in the domain controller cache.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for global catalog objects to 3600 seconds (60 minutes), which is the default value:

adclient.cache.expires.gc: 3600

> **Note**: If you do not define the adclient.cache.expires.gc parameter in the configuration file, it has a default value of 3600 seconds (1 hour). Unlike the default value for other object types, the default value for adclient.cache.expires.gc is not dependent on the value of adclient.cache.expires.

adclient.cache.expires.group

This configuration parameter specifies the number of seconds before a group object in the domain controller cache expires. The domain controller cache contains object attributes including the object's Active Directory properties, memberships, indexes and other parameters. If this parameter is not specified, the generic object cache expiration value is used.

Every group object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for group objects to 1800 seconds (30 minutes):

adclient.cache.expires.group: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.group parameter, its value overrides the default value for cached objects.

adclient.cache.expires.group.membership

This configuration parameter specifies the number of seconds before a group membership object in the domain controller cache expires. The domain controller cache contains object attributes including the object's Active Directory properties, memberships, indexes and other parameters. If this parameter is not specified, the generic object cache expiration value is used.

Every group membership object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for group objects to 1800 seconds (30 minutes):

adclient.cache.expires.group: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.group.membership parameter, its value overrides the default value for cached objects.

adclient.cache.expires.search

This configuration parameter specifies the number of seconds before the results of an Active Directory search expire.

Search expiration is handled separately from object expiration because a search result may include objects that have been deleted or be missing objects that have been added that meet the search criteria.

You can set this configuration parameter by manually adding it to the centrifydc.conf configuration file and specifying the maximum number of seconds for a search result to be kept in the local cache.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for search to 1800 seconds (30 minutes):

adclient.cache.expires.search: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.search parameter, its value overrides the default value for cached objects.

adclient.cache.expires.user

This configuration parameter specifies the number of seconds before a user object in the domain controller cache expires. If this parameter is not specified, the generic object cache expiration value is used.

Every user object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for user objects to 1800 seconds (30 minutes):

adclient.cache.expires.user: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.user parameter, its value overrides the default value for cached objects.

adclient.cache.expires.user.membership

This configuration parameter specifies the number of seconds before a user's group membership information in the domain controller cache expires. If this parameter is not specified, the user object cache expiration value (adclient.cache.expires.user) is used.

Every user object retrieved from Active Directory is stamped with the system time when it enters the domain controller cache. Once an object expires, if it is needed again, the agent contacts Active Directory to determine whether to retrieve an updated object (because the object has changed) or renew the expired object (because no changes have been made). To make this determination, the agent checks the highestUSN for the expired object. If the value has changed, the agent retrieves the updated object. If the highestUSN has not changed, the agent resets the object's timestamp to the new system time and retrieves the object from the cache.

If the agent is unable to contact Active Directory to check for updates to an expired object—for example because the computer is disconnected from the network—the agent returns the currently cached object until it can successfully contact Active Directory.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time for user objects to 1800 seconds (30 minutes):

adclient.cache.expires.user.membership: 1800

> **Note**: The default cache expiration time for all objects types is defined with the adclient.cache.expires parameter. If you explicitly set the adclient.cache.expires.user.membership parameter, its value overrides the default value for cache objects.

adclient.cache.flush.interval

This configuration parameter specifies how frequently, in hours, to flush all objects from the domain controller cache. The domain controller cache contains object attributes including the object's Active Directory properties, memberships, indexes and other parameters.The parameter value must be a positive integer. Unlike the other cache management parameters, which flush objects selectively, this parameter removes all objects in the cache at the interval you specify.

Specify the interval, in hours, using an integer value. The default value is 0, which disables the complete flushing of the cache.

For example, the following setting flushes all values in the cache every 12 hours:

adclient.cache.flush.interval: 12

adclient.cache.negative.lifetime

This configuration parameter specifies how long, in minutes, a negative object should remain in the domain controller cache. The domain controller cache contains object attributes including the object's Active Directory properties, memberships, indexes and other parameters. A negative object is returned when an object is not found in a search result. This configuration parameter determines how long that negative result should remain in the cache, regardless of the object type or object expiration time. By storing this negative result in the cache, the agent does not need to connect to Active Directory to look for an object that was previously not found.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be a positive integer. The default period of time for keeping negative results is 5 minutes. Setting the parameter value to 0 keeps negative objects in the cache indefinitely.

The following example sets the lifetime for negative objects to 10 minutes:

adclient.cache.negative.lifetime: 10

adclient.cache.object.lifetime

This configuration parameter specifies how long, in hours, an Active Directory object should remain in the domain controller cache. Setting the parameter value to 0 keeps objects in the cache indefinitely. When you set this parameter to 0, objects remain in the cache until they are deleted from Active Directory or the cache is manually flushed with the adflush command. If you don't want objects to remain in the cache indefinitely, you can use this parameter to set the maximum amount of time an object should be available in the cache.

For example, if you want to set the maximum time for an object to be held in the cache to 12 hours, you can set this configuration parameters as follows:

adclient.cache.object.lifetime: 12

With this setting, object values can be retrieved from the local domain controller cache for 12 hours. At the end of the 12 hour period, however, the object is removed from the local cache and must be retrieved from Active Directory if it is needed again.

If this parameter is not defined in the configuration file, its default value is 0.

adclient.cache.refresh

This configuration parameter specifies the maximum number of seconds an object can be read from the domain controller cache before it needs to be refreshed. This parameter allows an object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh: 5

> **Note**: This configuration parameter applies to generic objects in the domain controller cache and becomes the default refresh period for all object types. You can set separate refresh periods for specific objects types using the object-specific configuration parameters. For example, you can set different refresh times for computer objects and user objects using the adclient.cache.refresh.computer, and adclient.cache.refresh.user configuration parameters. This generic object refresh setting applies to any object for which you do not set an object-specific refresh period.

adclient.cache.refresh.computer

This configuration parameter specifies the maximum number of seconds a computer object can be read from the domain controller cache before it needs to be refreshed. This parameter allows a computer object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.computer: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.computer parameter, its value overrides the default value for objects.

adclient.cache.refresh.extension

This configuration parameter specifies the maximum number of seconds an extension object can be read from the domain controller cache before it needs to be refreshed. The domain controller cache contains object attributes including the object's Active Directory properties, memberships, indexes and other parameters. This parameter allows an extension object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.extension: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.extension parameter, its value overrides the default value for objects.

adclient.cache.refresh.gc

This configuration parameter specifies the maximum number of seconds an entry can be read from the global catalog cache before it needs to be refreshed. This parameter allows an object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.gc: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.gc parameter, its value overrides the default value for refreshing objects.

adclient.cache.refresh.group

This configuration parameter specifies the maximum number of seconds a group object can be read from the domain controller cache before it needs to be refreshed. This parameter allows a group object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.group: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.group parameter, its value overrides the default value for refreshing objects.

adclient.cache.refresh.search

This configuration parameter specifies the maximum number of seconds search results can be read from the domain controller cache before it needs to be refreshed. This parameter allows the search results to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.search: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.search parameter, its value overrides the default value for refreshing objects.

adclient.cache.refresh.user

This configuration parameter specifies the maximum number of seconds a user object can be read from the domain controller cache before it needs to be refreshed. This parameter allows a user object to be read from the cache if the age of the object in the cache is less than the parameter value.

This parameter is useful in cases where reading objects from Active Directory may result in duplicate object requests. For example, the PAM-enabled login process is designed to always retrieve the user object from Active Directory first to ensure that the most recent version of the user object is available for logging on. It only retrieves the user object from the cache if Active Directory is unavailable. Logging on, however, may require this same information to be requested from Active Directory more than once.

To prevent sending the duplicate object requests during the login process, the Delinea Agent checks this parameter. If the age of the object in the cache is less than the refresh time specified by this configuration parameter, the object is allowed to be read from cache. If the object in the cache is older than the refresh interval, the login process retrieves the information from Active Directory.

The parameter value must be a positive integer. The default value is 5 seconds. For example:

adclient.cache.refresh.user: 5

> **Note**: The default refresh time for all objects types is defined with the adclient.cache.refresh parameter. If you set the adclient.cache.refresh.user parameter, its value overrides the default value for refreshing objects.

adclient.cache.upn.index

This configuration parameter specifies whether to index user principle names (UPNs) that are stored in the Delinea user cache. You can use this parameter to differentiate between two users when the UPN of one user is equal to the SAM@domain_name of another user, and both user objects are stored in the user cache.

To enable UPN indexing, set this parameter to true. For example:

adclient.cache.upn.index: true

By default, this parameter is set to false, and UPNs are not indexed.

adclient.client.idle.timeout

This configuration parameter specifies the number of seconds before the agent will drop a socket connection to an inactive client.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be an integer greater than zero. The following example sets the inactive client timeout to 900 seconds:

adclient.client.idle.timeout: 900

If you set this parameter to zero, the agent will never drop the socket connection. Therefore, you should always specify a value greater than zero.

If this parameter is not defined in the configuration file, its default value is 5 seconds.

> **Note**: You must restart adclient for changes to this parameter to take effect. There is a Group Policy setting for this property but changing it has no effect until adclient is restarted on the affected computers. (Ref: CS-18792c)

adclient.clients.listen.backlog

This configuration parameter specifies the number of backlog connections to maintain when all threads are busy. Through operating system services, the agent maintains a queue of pending connection requests that are received from the processes that need the services of the agent. This configuration parameter controls the maximum number of pending requests to hold in the queue.

Decreasing the value of this parameter may prevent processes from performing tasks that require adclient services, for example, a login request may be unable to authenticate a user. Increasing the value of this parameter may reduce the chance of service request failure, but may waste system memory resources and impact system performance.

For example:

adclient.clients.listen.backlog: 50

If you change this parameter, you must restart the adclient process for the change take effect.

adclient.clients.socket

This configuration parameter specifies the named socket through which clients communicate with the agent.

The parameter value must be the name of the socket. For example:

adclient.clients.socket: /var/centrifydc/daemon

If this parameter is not defined in the configuration file, its default value is daemon.

adclient.clients.threads

This configuration parameter specifies the number of threads the agent pre-allocates for processing client requests.

The parameter value must be an integer zero or greater. If you set this parameter to zero, the agent processes requests sequentially. For example:

adclient.clients.threads: 4

If this parameter is not defined in the configuration file, its default value is 4 threads.

If you change this parameter, you must restart the adclient process for the change take effect.

adclient.clients.threads.max

This configuration parameter specifies the maximum number of threads the agent will allocate for processing client requests. This parameter value should be greater than or equal to the number of pre-allocated threads specified by the adclient.clients.threads parameter. The default value is 20 threads. For example:

adclient.clients.threads.max: 20

If you change this parameter, you must restart the adclient process for the change take effect.

adclient.clients.threads.poll

This configuration parameter specifies the number of milliseconds the agent waits between checks to see if a client's request has been completed.

Request completion polling is necessary to eliminate race conditions in operating environments such as Linux that don't have pselect implemented in the kernel. This polling mechanism should be disabled if the operating system has an atomic pselect.

The parameter value must be an integer zero or greater. A value of zero turns off polling. For example:

adclient.clients.threads.poll: 100

If this parameter is not defined in the configuration file, its default value is 100 milliseconds.

adclient.cloud.auth.token.max

This configuration parameter specifies the maximum number of cloud authentication requests that can be processed simultaneously. The default is 10 requests.

If you change this parameter, you must restart the adclient process. When the number of simultaneous connection requests exceeds this setting, the next authentication request will fail. If an authentication request times out waiting for a response, the connection is closed and the token is cleared to allow a new request

The default value of this parameter is 10 simultaneous requests. For example:

adclient.cloud.auth.token.max: 10

adclient.cloud.cert.store

Use this configuration parameter to specify the file in which to store the certificate that verifies cloud server connections.

By default, if there is no set file location, adclient will automatically locate the certificate.

> **Note**: The agent searches the following locations by default:

- /etc/ssl/certs/ca-certificates.crt
- /etc/pki/tls/certs/ca-bundle.crt
- /user/share/ssl/certs/ca-bundle.crt
- /usr/local/share/certs/ca-root-nss.crt
- /etc/ssl/cert.pem

This configuration parameter should only be used if adclient.cloud.skip.cert.verification.

For example:

adclient.cloud.cert.store: /etc/ssl/ca-bundle.crt

adclient.cloud.connector

This configuration parameter specifies a cloud connector to use in the current Active Directory forest. This parameter enables you to explicitly designate the cloud connector to use for connections between Delinea-managed Linux and UNIX computers and the cloud instance providing cloud authentication services.

By default, adclient will automatically select the most appropriate cloud connector to use based on network topology. You can use this parameter, however, to manually specify the IP address or fully-qualified domain name of the cloud connector you want connections to go through. By designating a cloud connector, adclient will use that cloud connector and won't do automatic discovery to other connectors.

For example, to specify the cloud connector by IP address:

adclient.cloud.connector: 192.168.1.61:8080

To specify the cloud connector using the fully-qualified domain name:

adclient.cloud.connector: connector.mydomain.com:8080

Note that port 8080 is the default port for cloud connectors to use.

adclient.cloud.connector.refresh.interval

This configuration parameter specifies how frequently adclient contacts its cloud connector. The refresh task is a background process that searches for and selects the nearest available cloud connector to use for connectivity between the Active Directory forest and the cloud service. By default, the process runs every 8 hours. You can use this group policy to modify that interval.

For example, to set the cloud connector refresh interval to 12 hours:

adclient.cloud.connector.refresh.interval: 12

adclient.cloud.skip.cert.verification

Use this configuration parameter to skip verification of the security certificate for cloud connections.

By default, this parameter is set to false, and certificate verification is required.

For example:

adclient.cloud.skip.cert.verification: false

adclient.cloud.connector.subnet.preference.enabled

This configuration parameter specifies whether or not to enable the ability to select subnet preferences when the agent connects to a cloud connector.

By default, this option is not enabled (false), which means that the agent selects the cloud connector based on the closest Active Directory site.

If you enable this option, the agent selects the cloud connector located in the same subnet as the client within the current Active Directory site, then in different subnets within the current Active Directory site, and then in an Active Directory site that's different than the current one.

adclient.custom.attributes

This configuration parameter enables you to add Active Directory attributes to the Delinea authentication cache that are not retrieved by default. You can specify computer, user, or group attributes by using the appropriate form of the parameter:

adclient.custom.attributes.**computer**:attributeNameadclient.custom.attributes.**user**:attributeNameadclient.custom.attributes.**group**:attributeName

Separate multiple attributes by a space. For example, to specify the user attributes comment and company and the group attributes info and telephoneNumber:

adclient.custom.attributes.user: comment company
adclient.custom.attributes.group: info telephoneNumber

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

> **Note**: You can use the adquery --dump command to see a list of the attributes that adclient caches for users or groups.

adclient.deploy.report.update.interval

The adclient.deploy.report.update.interval configuration parameter specifies how often adclient updates the postalAddress attribute of computer objects. You specify a number to represent how many hours the update interval is.

By default, this parameter is 1, so that the postalAddress is updated every hour.

Within a computer object's postalAddress are the following sub-attributes:

- CurrentDC (the current domain controller)
- AdclientProcessElapseTime (how long since adclient started, in seconds)
- ComputerUpTime (how long the computer has been up and running, "xxx days, hh:mm" format)
- DCUpdateTime (the timestamp when the domain controller was last updated with this information, in Windows UTC time format)
- CurrentConnector (the current connector in use)
- ConnectorUpdateTime (the timestamp when the connector was last updated with this information, in Windows UTC time format)

For example, to set the interval to 8 hours:

adclient.deploy.report.update.interval:8

adclient.disk.check.free

This configuration parameter specifies the size, in KB, of disk space available for the local cache that should generate a warning message. The agent will check the availability of free disk space at the interval specified with the adclient.disk.check.interval parameter. If the disk space available at any interval is less than the value you set for the adclient.disk.check.free parameter, the agent stops saving data in the local cache and displays a warning message to indicate that you should free up disk space. At the next interval when the available disk space exceeds the size you set for this parameter, the agent resumes normal operation and saving data to its cache.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an integer of zero or greater. A value of zero disables the display of a warning message about the available disk space. The default minimum of available disk space that triggers a warning message is 51200 KB. For example:

adclient.disk.check.free: 51200

> **Note**: Keep in mind that the value you set for this parameter can affect the recovery of a system. The agent will only resume saving data in its local cache if there is more disk space available than what you have specified to generate the warning. For example, if you have specified that the agent issue a warning when the available disk space falls to 51200 KB, there must be more than 51200 KB of disk space available for the agent to return to normal operation and write to the cache.

adclient.disk.check.interval

This configuration parameter specifies how frequently the agent should check the disk space available for the local cache. The default interval checks the available disk space every 5 minutes. If the disk space available at any interval is less than the value you set for the adclient.disk.check.free parameter, the agent will stop saving data in the local cache and will discard any new data until you free up enough disk space for it to resume saving data in the local cache.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an integer zero or greater. A value of zero disables checking for available disk space. For example:

adclient.disk.check.interval: 5

> **Note**: Keep in mind that the value you set for this parameter can affect the recovery of a system after the agent stops writing data to the local cache. If you set this parameter to 0, the agent will not check for available disk space so it will not return to normal operation when disk space is freed up. In addition, setting this parameter to 0 or to a long interval may cause the agent to consume too much of the disk for its local cache and make the computer unstable or unusable. Therefore, you should keep the interval for checking the available disk space relatively short. Keeping the interval short will also help to ensure that the agent resumes normal operation and saving data to its cache at the earliest opportunity.

adclient.dns.cache.timeout

> **Note**: This parameter has been deprecated in favor of dns.cache.timeout.

This configuration parameter specifies the amount of time, in seconds, before a cached DNS response expires.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be a positive integer. For example, the default values expires a cached DNS response after 300 seconds:

adclient.dns.cache.timeout: 300

adclient.dns.cachingserver

Cache-only DNS servers cannot provide sufficient authoritative responses to DNS requests directly. They refer to authoritative servers, such as a Windows server and then relay the answer to the DNS request. This means, for some cache-only DNS servers, DNS requests, sent to cache-only DNS server, need to have recursive flag. For example dnscache. Other cache-only DNS servers do not require setting the recursive flag. See your DNS server specifications.

Examples of cache-only DNS servers, include:

- dnsmasq
- dnscache
- tinyDNS
- pdnsd
- unbound
- dnrd

The adclient.dns.cachingserver configuration parameter determines whether to send recursive DNS requests or not. When set to true, this parameter sends recursive DNS requests, as apposed to the standard non-recursive requests. Default is false.

To use a cache-only DNS server, in the centrifydc.conf file, set in the adclient.dns.cachingserver parameter to true. There might be some DNS functionality loss in adclient, when this parameter is set to true.

Parameter syntax:

adclient.dns.cachingserver: false

The default setting is false.

When set to true, recursive DNS requests are allowed.

Optionally, the install.sh script also provides an option for handling cache-only DNS servers with adcheck.

install.sh [--dns_cache]

This invokes adcheck with option -r and allows DNS recursion with cache-only DNS servers.

adclient.dumpcore

This configuration parameter specifies whether the agent should be allowed to dump core.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The value you set for this parameter overrides the default ulimit setting. The parameter value must be one of the following valid options:

- never to specify that the agent never dump core.
- once to specify that the agent should dump core only when there is no existing core dump file.
- always to specify that the agent dump core on every crash.

For example:

adclient.dumpcore: never

adclient.dynamic.dns.command

This configuration parameter specifies the parameters to use for the addns command if it is launched by adclient (see adclient.dynamic.dns.enabled).

For example, the default setting is:

adclient.dynamic.dns.command: /usr/sbin/addns -U -m

The -U option creates or updates the IP address and domain name pointer (PTR) records in the DNS server for the local computer.

The -m option uses the local computer account's Active Directory credentials to establish a security context with the DNS server.

> **Note**: UNIX computers that act as a gateway between networks may require you to specify the network adapter IP address in the addns command line. To ensure that you register the correct network address with the Active Directory DNS server, set adclient.dynamic.dns.command with a command line that uses the correct IP address for the network interface you want to use. (Ref: CS-20319c)

adclient.dynamic.dns.enabled

This configuration parameter specifies whether adclient will automatically launch the addns command. The addns command dynamically updates DNS records on an Active Directory-based DNS server in environments where the DHCP server cannot update DNS records automatically.

> **Note**: In most cases, you do not need to use the addns command if a host's IP address is managed by a Windows-based DNS server and the host obtains its IP address from a Windows-based DHCP server because the DHCP server updates the DNS record for the host automatically. If you are not using a Windows-based DNS server, you should use nsupdate or a similar command appropriate to the operating environment of the DNS server to update DNS records.

The addns command is launched with the parameters specified by the adclient.dynamic.dns.command configuration parameter.

The default value for Mac OS X computers is True

The default value for all other platforms is False.

adclient.dynamic.dns.refresh.interval

This configuration parameter specifies whether or not dynamic DNS records are periodically updated for this host and, if there are updates, the interval between updates. The parameter takes an integer of 0 or greater. If set to 0, it turns the DNS update feature off. If set to 1 or greater, it specifies the number of seconds between DNS update attempts.

The default value for this parameter is 0 (off).

adclient.excluded.domains

This configuration parameter specifies a list of domains to exclude from the list of trusted domains.

For example, you might want to exclude specific domains that are contained within a trusted forest. To specify domains to exclude, enter one or more domain names in dotted-name format, separated by spaces, For example:

adclient.excluded.domains: eng.acme.com qa.acme.com

The Delinea Agent does not probe any excluded domains and consequently ignores users from these domains.

The default value for this parameter is the empty list, which does not exclude any domains.

adclient.exit.on.incomplete.zone.hierarchy

This configuration parameter specifies whether or not the agent stops if it can't load the complete zone hierarchy.

Adclient loads the zone hierarchy during startup only. There might be some situations where not all zones can be loaded: for example, if this is the first time loading zones since running adjoin. Normally, adclient will run with an incomplete or incorrect zone hierarchy.

The parameter value must be a boolean value of true or false. For example:

adclient.exit.on.incomplete.zone.hierarchy: true

If this parameter is not defined in the configuration file, its default value is false.

adclient.fetch.object.count

This configuration parameter specifies the number of objects to obtain in a single LDAP request. You can use this parameter to optimize the number of objects to suit your environment.

The parameter value must be a positive integer. For example:

adclient.fetch.object.count: 5

With this parameter, there is a trade-off here between speed and memory usage as well as bandwidth and latency. As you increase the number of objects included in an LDAP request, you may improve the overall performance by decreasing the number of connections to Active Directory and reducing the overall demand on the server, but you increase the RAM used by the agent. If you decrease the number of objects included in an LDAP request, you may reduce overall performance because of the additional network traffic, but decrease the memory used by the agent.

On faster networks, you can safely retrieve a small number of objects. On slower networks or when retrieving information for large groups (for example, groups with more than 1000 users), you may want to increase the value for this parameter.

adclient.force.salt.lookup

This configuration parameter specifies that you want to force the Delinea Agent to look up the complete principal name, including the Kerberos realm used as the key salt, from the KDC. Setting this parameter to true is required if you remove arcfour-hmac-md5 from the list of encryption types specified for the adclient.krb5.tkt.encryption.types parameter and if you change a userPrincipalName attribute in Active Directory without changing the user's password.

The parameter value can be true or false. The default value is true. For example:

adclient.force.salt.lookup: false

> **Note**: When this parameter value is set to true it may cause "pre-authrequired" warning messages to appear in the Active Directory event log.

adclient.gc.locator.shortcut

If set to true, the adclient.gc.locator.shortcut configuration parameter specifies to use a shortcut to the global catalog on the domain controller, as long as the domain controller is also a valid global catalog server.

This process bypasses the usual DNS lookup for the global catalog. Setting this parameter to true can be useful in situations where the root domain is blocked, thereby also blocking the global catalog.

By default, the adclient.gc.locator.shortcut configuration parameter is set to false.

adclient.get.primarygroup.membership

This configuration parameter specifies whether zone users are added as members of a primary group.

By default, Active Directory users are not members of their primary Active Directory group. This parameter is used to control whether zone users are added as members of this primary group.

The parameter value can be true or false. If you set this parameter to true, zone users are added to the primary group. If you set this parameter to false, zone users are not added to the primary group.

Setting this parameter to true can have performance implications when you query the group (for example, by using the adquery group command) because adclient has to search for all Active Directory users who have has this group as their primary group.

The default value is false. For example:

adclient.get.primarygroup.membership: false

adclient.gmsa

Use this configuration parameter to specify the gMSA (Microsoft group Managed Service Accounts on Windows) that adclient will treat either as Active Directory or Unix user accounts.

adclient.gmsa: **<gmsa>**

When you specify a gMSA, it is recommended to not use a field or format that uses special characters. Special characters have to be formatted with escape sequences and they're likely to cause errors. For example, if you use CN (CommonName), DisplayName, UPN (UserPrincipalName), those are fine, but samAccountName$ can be problematic because of its use of the $ character.

For each gMSA that you specify, you also need to specify the location where the password is stored using the following format:

<gmsa>.krb5.keytab: **<file_path>**

Example:

adclient.gmsa: serviceXYZ

serviceXYZ.krb5.keytab: /some/secure/location/serviceXYZ.keytab

adclient.hash.allow

This configuration parameter specifies the list of users you want to allow to have their password hash stored. By default, the Delinea Agent stores a UNIX-style SHA256 hash of each user's password in the cache when the user is authenticated during login. Storing the password hash allows previously authenticated users to log on when the computer is disconnected from the network or Active Directory is unavailable.

Although the default behavior is to store the password hash for all users, you can use this parameter to explicitly list the users whose hashed passwords are stored in the cache. If you use this parameter, only the users you specify can log on when the computer is disconnected from the network or Active Directory is unavailable.

The parameter value can be one or more user names. If more than one name, the names can be separated by commas or spaces. For example:

adclient.hash.allow: jdoe bsmith

If no user names are specified or the parameter is not defined in the configuration file, the password hash is stored for all users.

adclient.hash.deny

This configuration parameter specifies the list of users you want to prevent from having their password hash stored. By default, the Delinea Agent stores a UNIX-style SHA256 hash of each user's password in the cache when the user is authenticated during login. Storing the password hash allows previously authenticated users to log on when the computer is disconnected from the network or Active Directory is unavailable.

Although the default behavior is to store the password hash for all users, you can use this parameter to explicitly list the users whose hashed passwords must not be stored in the cache. If you use this parameter, the users you specify cannot log on when the computer is disconnected from the network or Active Directory is unavailable. All other users are permitted to have their password hash stored and allowed to log on when the computer is disconnected from the network or Active Directory is unavailable.

The parameter value can be one or more user names. If more than one name, the names can be separated by commas or spaces. For example:

adclient.hash.deny: jdoe bsmith

If no user names are specified or the parameter is not defined in the configuration file, the password hash is stored for all users.

adclient.hash.expires

This configuration parameter specifies the number of days the password hash for any user can be stored in the cache before it expires.

The parameter value must be a positive integer. A value of zero (0) specifies that the password hash should never expire. For example, to set this parameter so that the password hash expires after 7 days:

adclient.hash.expires: 7

If this parameter is not defined in the configuration file, its default value is 0.

adclient.heartbeat.interval

The adclient.heartbeat.interval configuration parameter specifies how often (in minutes) adclient will send an INFO message to the UNIX syslog.

By default, this parameter is set to zero (0), which means that this task is disabled.

For example, to set the interval to 5 minutes:

adclient.heartbeat.interval:5

adclient.ignore.setgrpsrc

This configuration parameter specifies whether adclient accesses the ~/home/.setgrpsrc file when a user issues commands such as groups and id that return information about users' group membership.

Whenever it accesses the ~/home/.setgrpsrc file, adclient mounts the specified user's home directory. This behavior can be problematic because it makes it difficult for administrators to control mounts to file servers. For example, moving a file server requires removing all mounts, so before doing so, an administrator must scan for mounts that may have been created by running groups or id. In addition, the extra NFS mounts generated by this behavior reduce the number of reserved ports available on Red Hat systems and can slow system performance.

You can set this parameter's value to either true or false. When true, adclient ignores the ~/home/.setgrpsrc file when a user issues commands such as groups and id, and does *not* mount the specified user's home directory. When false, adclient does check the ~/home/.setgrpsrc file when a user issues commands such as groups and id, and does mount the specified user's home directory.

The default value is false.

To change the value to true, such that adclient does not check the ~/home/.setgrpsrc file and mount a user's home directory when checking for group membership, set this parameter's value to true in the configuration file:

adclient.ignore.setgrpsrc: true

After setting this parameter, you must run adreload to reload the configuration file.

> **Note**: Setting this parameter does not affect the use of the adsetgroups command by the current user to view or change group membership. For a logged on user, the home directory is already mounted.

adclient.included.domains

This configuration parameter specifies a list of domains to include as trusted domains.

If this parameter specifies any domains (that is, is not empty), only the specified domains and the joined domain will be trusted. For example, you might want to specify specific domains to trust in a trusted forest, rather than trust all domains in the forest.

> **Note**: Alternately, you may use the adclient.excluded.domains parameter to exclude from the trusted list specific domains that are contained within a trusted forest.

To specify domains to include, enter one or more domain names in dotted-name format, separated by spaces, For example:

adclient.included.domains: eng.acme.com qa.acme.com

In this example, the only trusted domains are eng.acme.com, qa.acme.com, and the domain to which the computer is joined.

The Delinea Agent does not probe any domains that are not on the list (except the joined domain) and consequently ignores users from other domains.

The default value for this parameter is the empty list, which has no effect on determining which domains to trust.

adclient.ipv4.port.range.low / high

You can use the adclient.ipv4.port.range.low and adclient.ipv4.port.range.high parameters to specify the low and high ends of the range of IP ports for adclient and adnisd to use. These parameters control the outbound connection port for both TCP and UDP connections.

By specifying an IP port range, you can then configure your firewall to allow traffic through that port range only.

The typical port number is between 1024 and 65535. Setting this parameter does require a restart of adclient.

adclient.iterate.private.groups

This configuration parameter specifies whether adclient iterates through users to look for private groups when searching for groups.

The adclient process may receive periodic requests from processes such as adnisd for all zone-enabled users and groups. adclient queries Active Directory for those users and groups. By default, adclient queries only for group objects when searching for groups. When dynamic private groups are turned on (using the configuration parameter auto.schema.private.group), it creates private groups with a single user where the primary GID of the private group is set to the user's UID. When dynamic private groups are present, adclient must search through user objects as well as group objects when looking for groups.

This parameter's value must be either true or false. When true, adclient iterates through user objects in Active Directory when searching for groups. When false, adclient does not iterate through user objects when searching for groups.

Note that iterating through users isn't noticeably slower than iterating only through groups until the numbers of users get into tens or hundreds of thousands. In these numbers, iteration may take more time.

If this parameter is not defined in the configuration file, its default value is initially false. Once adclient encounters a private group, it sets this parameter's value to true for the rest of adclient's process lifetime or until a user sets this parameter in the configuration file.

adclient.krb5.principal.lower

This configuration parameter converts the principal in Kerberos tickets to lowercase for compatibility with some third-party applications.

Set to true to change the principal in Kerberos tickets to lowercase.

Set to false to leave the case unchanged for the principal in Kerberos tickets.

The default value is false.

> **Note**: Use this parameter when a machine is joined to classic zones or hierarchical zones. When a machine is in Auto Zone, please use auto.schema.name.lower instead.

adclient.krb5.conf.domain_realm.anysite

This configuration parameter specifies whether or not to search for all domain controllers in a kerberized realm or just the domain controllers within the current, preferred site.

If this parameter is set to true, then the system will list all reachable domain controllers in a kerberized realm, regardless of which site they're located in.

If this parameter is set to false, then only the domain controller in the current, preferred site is listed.

For example:

adclient.krb5.conf.domain_realm.anysite: true

If this parameter is not defined in the configuration file, its default value is false.

adclient.ldap.packet.encrypt

This configuration parameter specifies the LDAP encryption policy you use. For example, if your organization has a security policy that does not allow unencrypted LDAP traffic, you can use this parameter to specify that all connections to Active Directory are encrypted. If your organization isn't concerned with the encryption of LDAP data and you want better performance, you can force all connections to be unencrypted.

The parameter value must be one of the following valid options:

- Allowed to allow both encrypted and unencrypted LDAP traffic.
- Disabled to prevent encrypted LDAP traffic.
- SignOnly to require all LDAP traffic to be signed to ensure packet integrity, but not encrypted.
- Required to require all LDAP traffic to be signed and encrypted. If you select this setting and a server doesn't support encryption, the connection will be refused.

For example:

adclient.ldap.packet.encrypt: Allowed

If this parameter is not defined in the configuration file, its default value is Allowed.

adclient.ldap.socket.timeout

This configuration parameter specifies the time, in seconds, the agent will wait for a socket connection timeout during LDAP binding.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer greater than zero. For example:

adclient.ldap.socket.timeout: 30

If this parameter is not defined in the configuration file, its default value is 5 seconds.

adclient.ldap.timeout

This configuration parameter specifies the time, in seconds, the agent will wait for a response from Active Directory before it gives up on the LDAP connection during fetch, update, or delete requests.

If a request is made to Active Directory and a response is not received within the number of seconds specified by this parameter, that request is retried once. For the second request, the agent will wait up to twice as long for a response. If the second request is not answered within that amount of time, the connection to that specific domain controller is considered disconnected. For example, if you set this parameter value to 7, the agent waits 7 seconds for a response from Active Directory to a fetch, update, or delete request. If the request isn't answered within 7 seconds, the agent retries the request once more and waits up to 14 seconds for a response before switching to disconnected mode. This results in a maximum elapsed time of 21 seconds for the agent to determine that Active Directory is unavailable.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. For example:

adclient.ldap.timeout: 10

If this parameter is not defined in the configuration file, its default value is 7 seconds.

adclient.ldap.timeout.search

This configuration parameter specifies the time, in seconds, the agent will wait for a response from Active Directory before it gives up on the LDAP connection during search requests.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. For example:

adclient.ldap.timeout: 10

If this parameter is not defined in the configuration file, its default value is double the value specified for the adclient.ldap.timeout parameter.

adclient.ldap.trust.enabled

This configuration parameter specifies whether you want to allow the agent to query trusted domains and forests for transitive trust information. The parameter's value can be true or false. If you set this parameter to true, the adclient process generates a krb5.conf that includes information from all trusted forests and can be used to authenticate cross-forest users to Kerberos applications. If you set this parameter to false, the agent does not query external trusted domains or forests for information.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be true or false. The default value is true. For example:

adclient.ldap.trust.enabled: true

> **Note**: Querying external trusted forests can take a significant amount of time if the other forests are blocked by firewalls. You may want to set this parameter to false if your trust relationships, network topology, or firewalls are not configured properly for access.

adclient.ldap.trust.timeout

This configuration parameter specifies the maximum number of seconds to wait for responses from external forests and trusted domains when attempting to determine trust relationships. If your trusted domains and forests are widely distributed, have slow or unreliable network connections, or are protected by firewalls, you may want to increase the value for this parameter to allow time for the agent to collect information from external domains and forests.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be a positive integer. For example, to time out if a response is not received in within two minutes, you can set this parameter value to 120:

adclient.ldap.trust.timeout: 120

The default value is 5 seconds. Before changing this setting, you should consider your network topology, the reliability of network connections, and the network bandwidth, speed, and latency for connecting to external forests and domains. If the value is set too low to consistently receive a response, you may be unable to search trusted external domains.

adclient.legacyzone.mfa.background.fetch.interval

This configuration parameter specifies how often the Delinea Agent updates the cache with the list of groups in classic zones and Auto Zones specified in the following parameters:

adclient.legacyzone.mfa.required.groups

This is a background process that updates the cache periodically according to the interval specified. Enabling this configuration parameter will improve multi-factor authentication performance.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy, or to temporarily override group policy.

For example, to set the parameter interval to 45 minutes:

adclient.legacyzone.mfa.background.fetch.interval: 45

To disable this process, set the parameter value to 0.

The default parameter value is 30 minutes.

adclient.legacyzone.mfa.cloudurl

This configuration parameter specifies which cloud instance URL the agent will accesses in order to implement multi-factor authentication for users in classic zones and Auto Zones.

If all of the cloud connectors in the Active Directory forest use a single cloud instance URL, the agent will use this instance for multi-factor authentication by default, and you do not have to specify the URL using this parameter. If you have access to more than one cloud instance URL, you must specify the URL you would like use for multi-factor authentication for the zone using this parameter or the group policy that modifies this parameter.

If you have access to more than one cloud instance URL, but do not specify which one should be used for multi-factor authentication, you will not be able to configure the zone to use multi-factor authentication.

In most cases, you set this configuration parameter using group policy. If you are manually setting this parameter, the parameter value must be a URL in the following format:

https://tenantid.domainfqdn:port/

For example:

adclient.legacyzone.mfa.cloudurl: https://abc0123.mydomain.com:8080/

adclient.legacyzone.mfa.enabled

This configuration parameter specifies whether multi-factor authentication is enabled for a classic zone or an Auto Zone. If you enable multi-factor authentication, users must be authenticated using more than one method. For example, users might be required to provide a password and respond to a text message or a phone call, or answer a security question. To enable multi-factor authentication, set this parameter to true. Set the parameter to false if multi-factor authentication is not required for any users.

In most cases, you set this configuration parameter using group policy. If you are manually setting this parameter, the parameter value must true or false. For example:

adclient.legacyzone.mfa.enable: true

If this parameter is not defined in the configuration file, its default value is false.

adclient.legacyzone.mfa.required.groups

This configuration parameter specifies a list of Active Directory groups in a classic zone or an Auto Zone that are required to use multi-factor authentication when logging on or using privileged commands. For example, if you want to require all members of the Qualtrak Admin group to use multi-factor authentication when they log on to computers that host sensitive information, you can add that group to this parameter.

Groups specified in this parameter must be security groups; distribution groups are not supported.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy, or to temporarily override group policy.

By default, multi-factor authentication is not enabled for groups in classic or Auto Zones.

You can separate each group by a space or a comma and you can use double quotes or escape characters to included spaces or special characters in group names. For example:

adclient.legacyzone.mfa.required.groups: centrify_users, “Qualtrak Admins”, Domain\ Users

Supported group name formats

You can specify groups by name or you can list the group names in a file in the following formats:

- SAM account name: sAMAccountName
- SAM account name of a group in a different domain: sAMAccountName@domain
- canonicalName: domain/container/cn

Specifying the parameter value in a separate file

To specify a file that contains a list of Active Directory group names, you can set the parameter value using the file: keyword and a file location. For example:

adclient.legacyzone.mfa.required.groups: file:/etc/centrifydc/legacy_user_groups_mfa.require

In the etc/centrifydc/legacy_user_groups_mfa.require file, you would type each group name on its own line using any of the supported name formats. For example:

server_users
“Qualtrak Admins”
Domain\ Users
group4@domain.com

adclient.legacyzone.mfa.required.users

This configuration parameter specifies a list of Active Directory users in a classic zone or an Auto Zone that are required to use multi-factor authentication when logging on or using privileged commands. For example, if you want to require Bill Hill to use multi-factor authentication to log on to a computer that hosts sensitive information, you can add her to this parameter.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy, or to temporarily override group policy.

By default, multi-factor authentication is not enabled for users in classic or Auto zones.

You can separate each user name by a space or a comma and you can use double quotes or escape characters to include spaces or special characters in user names.

For example, to specify that multi-factor authentication is required for users bill hill and tetsu.xu to log on to computers in an Auto Zone you would define the parameter value in the following way:

adclient.legacyzone.mfa.required.users: “bill.hill”, tetsu.xu@ajax.org

Supported user name formats

You can specify users by name or you can list the user names in a file in the following formats:

- SAM account name: sAMAccountName
- SAM account name of a user in a different domain: sAMAccountName@domain
- User Principal Name: name@domain
- Canonical Name: domain/container/cn
- Full DN: CN=commonName,...,DC=domain_component,DC=domain_component
- An asterisk (*), which includes all Active Directory users.

Specifying the parameter value in a separate file

To specify a file that contains a list of Active Directory user names, you can set the parameter value using the file: keyword and a file location. For example:

adclient.legacyzone.mfa.required.users: file:/etc/centrifydc/legacy_user_users_mfa.require

In the etc/centrifydc/legacy_user_users_mfa.require file, you would type each user name on its own line using any of the supported name formats. For example:

tetsu.xu
jane/ doe@ajax.org
Domain Users

adclient.legacyzone.mfa.rescue.users

This configuration parameter specifies a list of Active Directory users who can log on to computers in a classic zone or an Auto Zone when multi-factor authentication is required, but the agent cannot connect to the Delinea cloud service. You should specify at least one user account for this parameter to ensure that someone can access the computers in the event that multi-factor authentication is required but not available.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy, or to temporarily override group policy.

You can separate each user by a space or a comma and you can use double quotes or escape characters to include spaces or special characters in user names.

For example, to specify that user amy adams has the ability to log on to a computer in an Auto Zone if she is required but unable to authenticate using multi-factor authentication, you would define the parameter value in the following way:

adclient.legacyzone.mfa.rescue.users: amy.adams

Supported user name formats

You can specify users by name or you can list the user names in a file in the following formats:

- SAM account name: sAMAccountName
- SAM account name of a user in a different domain: sAMAccountName@domain
- User Principal Name: name@domain
- Canonical Name: domain/container/cn
- Full DN: CN=commonName,...,DC=domain_component,DC=domain_component
- An asterisk (*), which includes all Active Directory users.

Specifying the parameter value in a separate file

To specify a file that contains a list of Active Directory user names, you can set the parameter value using the file: keyword and a file location. For example:

adclient.legacyzone.mfa.rescue.users: file:/etc/centrifydc/legacy_user_users_mfa.rescue

In the etc/centrifydc/legacy_user_users_mfa.rescue file, you would type each user name on its own line using any of the supported name formats. For example:

tetsu.xu
amyadams
jane/ doe@ajax.org
user1@domain.com

adclient.legacyzone.mfa.tenantid

This configuration parameter specifies which Delinea Platform instance ID (also known as the tenant ID) the agent will access in order to implement multi-factor authentication for users in classic zones and Auto Zones.

By default, this parameter is empty and the agent uses the adclient.legacyzone.mfa.cloudurl parameter to locate Delinea Connectors.

For example:

adclient.legacyzone.mfa.tenantid: ABC1234

adclient.local.account.manage

This configuration parameter specifies whether the agent manages local user and local group accounts on computers where the agent is installed.

When this parameter is set to true:

- The agent gets the local user and local group profiles from the zone, and updates the local /etc/passwd and /etc/group files using the information defined in the zone.
- You can view and manage local users and groups in Access Manager as described in the *System Administrator's Guide for Linux and UNIX*.

The default value of this parameter is false, unless you upgraded from an Server Suite release in which local account management was enabled (in which case, it is set to true).

You can also set this configuration parameter using group policy.

adclient.local.account.manage.strict

This configuration parameter applies enforcement mode for local account management. The default is false and it is defined as not strict.

The following are sub-parameters for this configuration parameter:

- adclient.local.account.manage.strict.passwd: false
- adclient.local.account.manage.strict.group: false

When enabled in strict mode for user (except user with UID 0) any unmanaged local user's password entry is removed from /etc/passwd. If /etc/shadow file exist, shadow entry is removed as well. If user's extended attributes exist, those are removed.

When enabled in strict mode for group (except user with GID 0), any unmanaged local group entry is removed from /etc/group. If group's extended attributes exist, those are removed as well.

After switching to strict enforcement of local account management, switching back to non strict local account management does not restore the unmanaged local user or group.

adclient.local.account.notification.cli

This configuration parameter lets you define a command to process changes to local account profiles after the agent synchronizes local user and group profiles with profiles defined in a zone.

For example, if new local users are added, removed, or have their enabled/disabled status changed locally, the command that you define in this parameter is executed. Typical activities that this command might perform include setting the password for new or updated local accounts, or notifying password vault about changes to local accounts and defining actions to take regarding those accounts.

When this parameter is enabled, the agent invokes the defined command in another process and passes a comma separated UNIX name list to the command for further processing.

By default, this parameter's value is empty (that is, no command is defined). This parameter takes effect only when local account management is enable through group policy, or when the adclient.local.account.manage parameter is set to true.

You can also set this configuration parameter using group policy.

adclient.local.account.notification.cli.arg.length.max

This configuration parameter specifies the maximum argument length for the command that you define in the adclient.local.account.notification.cli parameter.

To determine the default argument length for your environment, execute getconf ARG_MAX.

After determining the default argument length for your environment, you can set this parameter to the same value to ensure that the agent's setting is consistent with the environment setting.

The default value of this parameter is 131072, which is 128KB. This parameter takes effect only when the adclient.local.account.manage parameter is set to true.

adclient.local.forest.altupn.lookup

This configuration parameter specifies whether or not to perform the local forest altupn lookup. The default is true. If you set this parameter to false, the local forest altupn lookup is skipped.

adclient.local.group.merge

This configuration parameter determines whether to merge local group membership from the /etc/group file into the zone group membership for groups that have the same name and GID. For example, if the Delinea Agent retrieves the membership list of kwan, emily, and sam for the group profile with the group name performx1 and GID 92531 from Active Directory and there is also a local group named performx1 with the GID 92531 with users wilson and jae, the merged group would include all five members (kwan,emily,sam,wilson,jae).

By default, this parameter value is set to false to prevent unexpected results. For example:

adclient.local.group.merge: false

Setting this parameter to true violates normal NSS behavior and, therefore, may have unexpected side effects. You should analyze your environment carefully before changing this parameter to true. If you determine you can safely merge local and Active Directory group profiles, you can uncomment this parameter and change its value.

> **Note**: If you set this parameter to true, you must run adreload to detect changes in the local group file.

adclient.logonhours.local.enforcement

This configuration parameter determines whether the agent and Active Directory both check for user logon hour restrictions, or whether only Active Directory checks for logon hour restrictions. This parameter is useful in cases where users are in time zones that are different from the time zone that the agent is in.

When this parameter is set to true, the agent and Active Directory both check for local logon hour restrictions. If the agent and user are in different time zones, and one time zone recognizes Daylight Savings Time while the other does not, the user may not be able to log on during permissible hours. (Ref: CS-33553 a)

When this parameter is set to false, only Active Directory checks for local hour restrictions, so there is no Daylight Savings Time conflict with the agent.

The default value for this parameter is true.

You should set this parameter to false if you have users that are not in the same time zone as the agent.

For example:

adclient.logonhours.local.enforcement: false

adclient.lookup.sites

This configuration parameter specifies a list of sites, and optionally a domain, to search for domain controllers and the global catalog if they are not found in the preferred site.

> **Note**: You can specify the preferred site in the adclient.preferred.site configuration parameter, and the preferred site is displayed when you execute the adinfo command.

The format for this parameter is:

adclient.lookup.sites: site1 [site2] [site3]...

The agent performs the following steps whenever it attempts to connect to a DC or GC:

1. Discover the preferred site.
2. From DNS, get a list of DCs or GCs in the preferred site and attempt to connect to each one until a connection is successful or the list is exhausted.
3. If unable to connect to a DC or GC in the preferred site, try to connect to a DC or GC in any site.

By using this configuration parameter, you can restrict step 3 to a specific set of alternate sites to search for DCs or GCs. Run Active Directory Sites and Services to see a list of sites for your environment. Sites are searched in the list order that you specify.

You can use any legal Active Directory site name when you set this parameter. For example:

adclient.lookup.sites USTEXAS USCALIFORNIA

You can optionally specify a domain suffix in this parameter, so that the site list is searched only in the domain that you specify. Use the following format to specify a domain:

adclient.lookup.sites.domainsuffix: site1 [site2] [site3]...

For example:

adclient.lookup.sites.example.com: USTEXAS USCALIFORNIA

If this configuration parameter is not configured, the agent tries to connect to a DC or GC in any site, as described in step 3 above. By default, this configuration parameter is not configured.

> **Note**: Do not add the preferred site to this list, as the preferred site will be searched anyway.

adclient.lrpc2.receive.timeout

This configuration parameter specifies how long, in seconds, the agent should wait to receive data coming from a client request.

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. For example:

adclient.lrpc2.receive.timeout: 30

If this parameter is not defined in the configuration file, its default value is 30 seconds.

adclient.lrpc2.send.timeout

This configuration parameter specifies the maximum number of seconds the agent should wait for reply data to be sent in response to a client request.

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. For example:

adclient.lrpc2.send.timeout: 30

If this parameter is not defined in the configuration file, its default value is 30 seconds.

adclient.next.closest.site.lookup.enabled

The adclient.next.closest.site.lookup.enabled configuration parameter specifies whether or not the Microsoft "Try Next Closest Site" is enabled. If it is enabled, the agent locates a domain controller according to the following order:

1. Tries to find a domain controller in the same site.
2. If there isn't a domain controller in the same site, the agent tries to find a domain controller in the next closest site.
   A site is closer if it has a lower site-link cost than another site with a higher site-link cost.
3. If no domain controller is available in the next closest site, the agent tries to find a domain controller in the domain.

By default, this parameter is set to true.

For example: adclient.next.closest.site.lookup.enabled:true.

adclient.nss.statistic.interval

The adclient.nss.statistic.interval configuration parameter specifies how much time passes before adclient updates the NSS query statistics information. The value represents the number of minutes.

By default, this parameter is set to 30 minutes.

For example: adclient.nss.statistic.interval:30.

adclient.ntlm.domains

This configuration parameter allows you to manually map Active Directory domain names to NTLM domains. This parameter is useful in cases where you need to use NTLM authentication because firewalls prevent Kerberos authentication and when firewall constraints prevent the automatic discovery of Active Directory to NTLM domain mapping.

You can specify the parameter's value as one or more domain name pairs, separated by a colon (:), or using the file: keyword and a file location. For example, you can set the parameter value using the format ActiveDirectory_DomainName:NTLM_DomainName to specify a list of domain name pairs:

adclient.ntlm.domains: AJAX.ORG:AJAX FIREFLY.COM:FIREFLY

To specify a file that contains a list of colon-separated values in the form of ActiveDirectory_DomainName:NTLM_DomainName, you can set the parameter value using the file: keyword and a file location:

adclient.ntlm.domains: file:/etc/centrifydc/domains.conf

Keep in mind that you must manually define how Active Directory domains map to NTML domains. If you define this information in a separate file, such as domains.conf, the file should consist of entries similar to the following:

AJAX.ORG:AJAX

FIREFLY.COM:FIREFLY

HR1.FIREFLY.COM:HR1

After you have manually defined the mapping of Active Directory domains to NTLM domains, you can use the pam.ntlm.auth.domains parameter to specify the list of domains that should use NTLM authentication instead of Kerberos authentication. For more information about defining the domains that should use Kerberos authentication, see pam.ntlm.auth.domains.

Alternatively, you can set the group policy, **Computer Configuration >Delinea Settings > DirectControl Settings > Network and Cache Settings > Specify AD to NTLM domain mappings**.

adclient.ntlm.separators

This configuration parameter specifies the separators that may be used between the domain name and the user name when NTLM format is used. For example, the following setting:

adclient.ntlm.separators: +/\\

allows any of the following formats (assuming a user joe in the acme.com domain):

acme.com+joe
acme.com/joe
acme.com\joe

> **Note**: The backslash character (\) can be problematic on some UNIX shells, in which case you may need to specify domain\\user.

The first character in the list is the one that adclient uses when generating NTLM names.

The default values are +/\\, with + being the adclient default.

adclient.one-way.x-forest.trust.force

Use this configuration parameter, adclient.one-way.x-forest.trust.force, to specify a list of two-way trusted domains that need to be treated as one-way trusted domains. This is useful when two-way trusted domains are not accessible from currently joining machine, for example, they are behind a firewall. Configuring this parameter allows x-forest users to authenticate onto the trusting machines.

The options are:

- An empty list (default)
- A list of forests or domains to be treated as one-way trusted.

  Specify a list of two-way trusted forests, and domains that have two-way external trust relationship with the local domain, to be treated by DirectControl Agent as one-way trusted forests or domains.

This parameter is likely to be used together with the configuration parameters, pam.ntlm.auth.domains and adclient.ntlm.domains, if these forests and domains are not accessible from the currently joining machine.

- Use the pam.ntlm.auth.domains parameter to specify the list of domains that use NTLM authentication instead of Kerberos authentication.
- Use the adclient.ntlm.domains parameter to map AD domains to NTLM domains.

Alternatively, you can set the group policy **Computer Configuration > Centrify Settings > DirectControl Settings > Adclient Settings > Force domains and forests to be one-way trusted**.

adclient.os.name

This configuration parameter specifies the name of the operating system for the local computer. This information is dynamically determined by the uname command and stored in the Active Directory computer object. The configuration parameter value can be manually overridden by defining a different value, if desired. If you change the value, however, you must restart adclient for the change to take effect.

For example, to set the parameter value manually to Linux:

adclient.os.name: Linux

adclient.os.version

This configuration parameter specifies the version of the operating system for the local computer. This information is dynamically determined by the uname command and stored in the Active Directory computer object. The configuration parameter value can be manually overridden by defining a different value, if desired. If you change the value, however, you must restart adclient for the change to take effect.

For example, to set the parameter value manually to 3.0-125:

adclient.os.version: 3.0-125

adclient.os.version.use.win7prefix

This configuration parameter specifies whether the operating system version prefix (6.1:) should be added automatically to the computer object's operatingSystemVersion attribute when a computer joins the domain. This prefix is used to indicate whether a computer supports FIPS encryption. The valid values are:

- 1 to add the prefix only when FIPS encryption is enabled.
- 2 to add the prefix regardless of FIPS encryption.

Depending on the version of the agent you have installed, the default value might be either of these values. The recommended setting for this parameter is 2. For example, to always add the prefix:

adclient.os.version.use.win7prefix: 2

adclient.paged.search.max

This configuration parameter specifies the maximum number of items included in each page of a paged LDAP search.

The parameter value must be a positive integer. For example:

adclient.paged.search.max: 100

If this parameter is not defined in the configuration file, its default value is 100 items.

Before changing this parameter setting, you should consider its impact on your environment. As you decrease the number of items included in each LDAP page, you increase the number of connections made to Active Directory and the added demand that increased traffic places on the server, but you decrease the RAM used by the agent. If you increase the number of items included in each LDAP page, you decrease the number of connections to Active Directory and reduce the overall demand on the server, but you increase the RAM used by the agent.

adclient.prefer.cache.validation

This configuration parameter instructs adclient to authenticate the user using the cached credentials first regardless of the current connectivity state with the Active Directory domain controller.

The parameter value is either true or false. The default is false; for example

adclient.prefer.cache.validation: false

Set this option to true to reduce traffic on slow networks. However, if the Active Directory credentials are not synchronized with the cached credentials, you run the risk of undesired side affects when the computer is online.

You can also set this configuration parameter using group policy.

adclient.preferred.login.domains

This configuration parameter enables you to specify the domain names against which to authenticate SAM account names. Use this parameter if your environment contains identical SAM account names on multiple domains, and you want to authenticate against a specific domain.

If you use this parameter, the adclient.cache.upn.index parameter must be set to true.

To use this parameter, type a space-separated list of domains as the parameter value. For example:

adclient.preferred.login.domains: demo1.acme.com demo2.acme.com

adclient.preferred.site

This configuration parameter enables you to identify a specific site to use to locate available domain controllers. By default, the adclient process uses CLDAP NETLOGON requests to automatically discover its site based on how sites are configured using Active Directory Sites and Services. This default behavior enables adclient to select domain controllers in the same site as preferred domain controllers because they are likely to provide the best performance and least replication delays. This configuration parameters enables you to override the site returned by Active Directory and use a specific site.

If you don't define a value for the parameter, adclient continues to discover sites based on how sites are configured using Active Directory Sites and Services.

If you want to define a specific site to use, you can use the following override options:

- You can specify a "universal" site override that does not include an Active Directory forest in the parameter name. The override applies to all Active Directory domains that do not have a forest-specific override.
- You can specify one or more "forest-specific" site overrides that includes the name of an Active Directory forest in the parameter names. This type of override limits the domain controllers to the domain controllers in the specified forest-specific site.

Forest-specific site overrides take precedence over universal site overrides. Depending on your requirements, you can use the site override options to override sites for all forests, specific forests, or a combination of the two.

The following is an example a "universal" site override that applies to all forests that do not have a forest specific override:

adclient.preferred.site: my-preferred-site

To specify a forest-specific site override, you specify the configuration parameter using the following format:

adclient.preferred.site.forest_name: my-forest-site

The following example illustrates how you would define the configuration parameters to use the USNORTH Active Directory site for all forests except the ocean-site forestspecific site.

adclient.preferred.site: USNORTH
adclient.preferred.site.ocean.net: ocean-site

adclient.prevalidate.allow.groups

This configuration parameter specifies the groups that are prevalidated to access the local UNIX computer using Active Directory credentials when the computer is offline even if users in the group have not previously logged onto the computer.

Under normal circumstances, only users who have previously logged on to a computer can be authenticated when the computer is disconnected from the network. For those users, authentication is based on the password hashes stored during the previous log-on. In some cases, however, you may require users who have never logged on to a particular computer to be authenticated when the computer is disconnected from the network. For example, you may have an administrative group that requires access to computers that are disconnected from the network but on which they have never previously logged in. To accommodate the users in that group, you can configure the group for prevalidation.

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a comma-separated list of UNIX group names. Enclose group names with spaces in double quotes, for example:

adclient.prevalidate.allow.groups: performx,qualtrak,"domain admins

Using this parameter with other prevalidation parameters

If you do not specify any groups for this parameter, then no group accounts are prevalidated to access the local computer. If you specify either the adclient.prevalidate.allow.users or adclient.prevalidate.allow.groups parameters, only those users and groups are prevalidated, with the exception of any users or groups specified by adclient.prevalidate.deny.users and adclient.prevalidate.deny.groups parameters. For example, to allow all users in the admins group to be prevalidated, except the users who are also members of the outsource group, you could set the adclient.prevalidate.allow.groups and adclient.prevalidate.deny.groups parameters like this:

adclient.prevalidate.allow.groups: admins
adclient.prevalidate.deny.groups: outsource

To allow prevalidation for all users in the zone without any exceptions, you can set the adclient.prevalidate.allow.groups parameter to all@zone. For example:

adclient.prevalidate.allow.groups: all@zone

For users or groups of users to be prevalidated, their accounts must be active accounts with permission to log on to the local computer and have a service principal name (SPN) set in the form of:

preval/username

Where preval is the service name specified by the adclient.prevalidate.service parameter and username is the user logon name, which can be either of the following:

- the name part of the user's UPN, if the domain part matches the user's domain
- samAccountName, if the UPN is empty or the UPN's domain part is different from the user's domain

Registering service principal names

To enable prevalidation for a user, you can use the Windows setspn.exe utility to add a service principal name for the user. For example, to register the service principal name for the user kai@arcade.com using preval as the service name, you could type a command similar to the following in a Windows Command Prompt window:

setspn -A preval/kai kai

This setspn command registers the SPN in Active Directory for the preval service for the specified user account, the Active Directory user kai. On the computers where this user is allowed to be prevalidated, the user can be authenticated without having logged on previously.

If you are allowing prevalidation for an administrative group, you must register a service principal name (SPN) for each member of the group. For example, if you are allowing prevalidation for the admins group and this group has five members, you would use the setspn.exe utility to register a Service Principal Name for each of those members.

Specifying the supported encryption types

All prevalidated users must have their Active Directory msDSSupportedEncryptionTypes attribute set to 0x18 (for just AES128 and AES256 support) or above to be able to login when disconnected. The parameter value represents the sum of the encryption types supported. Use the sum of the following encryption type values to determine the parameter value:

DES_CBC_CRC = 0x01
DES_CBC_MD5= 0x02
RC4_HMAC_MD5 = 0x4
AES128_CTS_HMAC_SHA1_96 = 0x08
AES256_CTS_HMAC_SHA1_96 = 0x10

For example, 0x1c indicates support for RC4_HMAC-MD5, AES128_CTS_HMAC_SHA1_96, and AES256_CTS_HMAC_SHA1_96.

Refreshing prevalidated credentials

To ensure their validity, the credentials for prevalidated users and groups are periodically retrieved from Active Directory. For example, the credentials are refreshed whenever you do the following:

- Reboot the local computer.
- Start or restart the adclient process.
- Run the adflush command to clear the cache.
- Changes a password from the local system.

The credentials are also periodically refreshed at the interval defined by the adclient.prevalidate.interval parameter to ensure that prevalidation will continue working after password changes.

adclient.prevalidate.allow.users

This configuration parameter specifies the users that are prevalidated to access the local UNIX computer using Active Directory credentials when the computer is offline even if they have not previously logged onto the computer.

Under normal circumstances, only users who have previously logged on to a computer can be authenticated when the computer is disconnected from the network. For those users, authentication is based on the password hashes stored during a previous log on. In some cases, however, you may require users who have never logged on to a particular computer to be authenticated when the computer is disconnected from the network. For example, you may have administrative users who require access to computers that are disconnected from the network but on which they have never previously logged in. To accommodate those users, you can configure them for prevalidation.

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a comma-separated list of UNIX user names. Enclose user names with spaces in double quotes, for example:

adclient.prevalidate.allow.users: jesse,rae,tai,"sp1 user"

sing this parameter with other prevalidation parameters

If you do not specify any users for this parameter, then no specific user accounts are prevalidated to access the local computer. If you specify either the adclient.prevalidate.allow.users or adclient.prevalidate.allow.groups parameters, only those users and groups are prevalidated, with the exception of any users or groups specified by adclient.prevalidate.deny.users and adclient.prevalidate.deny.groups parameters. For example, to allow all users in the admins group and the users ali, kai, and tanya who are not members of the admins group to be prevalidated, but prevent the users jorge and maurice from being prevalidated, you could set the allow and deny parameters like this:

adclient.prevalidate.allow.groups: admins
adclient.prevalidate.allow.users: ali,kai,tanya
adclient.prevalidate.deny.users: jorge,maurice

For users or groups to be prevalidated, their accounts must be active accounts with permission to log on to the local computer and have a Service Principal Name (SPN) set in the form of:

preval/username

Where preval is the service name specified by the adclient.prevalidate.service parameter and username is the user logon name, which can be either of the following:

- the name part of the user's UPN, if the domain part matches the user's domain
- samAccountName, if the UPN is empty or the UPN's domain part is different from the user's domain

Registering service principal names

To enable prevalidation for a user, you can use the Windows setspn.exe utility to add a service principal name for the user. For example, to register the Service Principal Name for the user kai@arcade.com using preval as the service name, you could type a command similar to the following in a Windows Command Prompt window:

setspn -A preval/kai kai

This setspn command registers the SPN in Active Directory for the preval service for the specified user account, the Active Directory user kai. On the computers where this user is allowed to be prevalidated, the user can be authenticated without having logged on previously.

Specifying the supported encryption types

All prevalidated users must have their Active Directory msDSSupportedEncryptionTypes attribute set to 0x18 (for just AES128 and AES256 support) or above to be able to login when disconnected. The parameter value represents the sum of the encryption types supported. Use the sum of the following encryption type values to determine the parameter value:

DES_CBC_CRC = 0x01
DES_CBC_MD5= 0x02
RC4_HMAC_MD5 = 0x4

AES128_CTS_HMAC_SHA1_96 = 0x08
AES256_CTS_HMAC_SHA1_96 = 0x10

For example, 0x1c indicates support for RC4_HMAC-MD5, AES128_CTS_HMAC_SHA1_96, and AES256_CTS_HMAC_SHA1_96.

Refreshing prevalidated credentials

To ensure their validity, the credentials for prevalidated users and groups are periodically retrieved from Active Directory. For example, the credentials are refreshed whenever you do the following:

- Reboot the local computer.
- Start or restart the adclient process.
- Run the adflush command to clear the cache.
- Changes a password from the local system.

The credentials are also periodically refreshed at the interval defined by the adclient.prevalidate.interval parameter to ensure that prevalidation will continue working after password changes.

adclient.prevalidate.deny.groups

This configuration parameter specifies the groups that cannot be prevalidated to access the local UNIX computer. If you allow any groups or users to be prevalidated, you can use this parameter to define exceptions for any groups that should be prevented from prevalidation. In most cases, you would use this parameter to exclude a subset of users that are in a member group of an allowed group. For example, to allow all users in the admins group to be prevalidated, except the users who are members of the outsource subgroup, you could set the adclient.prevalidate.allow.groups and adclient.prevalidate.deny.groups parameters like this:

adclient.prevalidate.allow.groups: admins
adclient.prevalidate.deny.groups: outsource

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a comma-separated list of UNIX group names. Enclose group names with spaces in double quotes, for example:

adclient.prevalidate.deny.groups: performx,qualtrak,"domain admins"

adclient.prevalidate.deny.users

This configuration parameter specifies the users that cannot be prevalidated to access the local UNIX computer. If you allow any groups or users to be prevalidated, you can use this parameter to define exceptions for any users who should be prevented from prevalidation. In most cases, you would use this parameter to exclude a subset of users that are members of an allowed group. For example, to allow all users in the admins group except the users jorge and maurice who are members of the admins group to be prevalidated, you could set the allow and deny parameters like this:

adclient.prevalidate.allow.groups: admins
adclient.prevalidate.deny.users: jorge,maurice

In most cases, you set this configuration parameter using group policy.

If you are manually setting this parameter, the parameter value must be a comma-separated list of UNIX user names. Enclose user names with spaces in double quotes, for example:

adclient.prevalidate.deny.users: jesse,rae,tai,"sp1 user"

adclient.prevalidate.interval

This configuration parameter specifies the interval, in hours, for refreshing the credentials for prevalidated user and group accounts. The credentials for prevalidated users must be periodically refreshed to ensure they are in sync with Active Directory and that prevalidation will continue working after password changes.

The parameter value should be a positive integer. A value of 0 disables all prevalidation of users. For example, to refresh the credentials for prevalidated users every 8 hours:

adclient.prevalidate.interval: 8

In most cases, you set this configuration parameter using group policy.

adclient.prevalidate.service

This configuration parameter specifies the service name to use for prevalidated users and groups. You must use the name you specify in this parameter when you register the Service Principal Name (SPN) for a user or group with the setspn.exe utility.

For example, to set the service name to preval:

adclient.prevalidate.service: preval

In most cases, you set this configuration parameter using group policy.

adclient.random.password.generate.try

This configuration parameter specifies the maximum number of times that the agent attempts to generate a random Active Directory password. Depending on the complexity requirements of your environment, you may need to set this value higher than the default to ensure an appropriately complex password is generated.

The default value is 10.

For example:

adclient.random.password.generate.try: 10

adclient.random.password.complexity.pattern

This configuration parameter specifies the complexity requirements for the generation of a random Active Directory password. Each complexity requirement is assigned a numeric value:

English uppercase characters (A through Z) = **1**

English lowercase characters (a through z) = **2**

Base 10 digits (0 through 9) = **4**

Special, non-alphanumeric characters (!, $, #, %, etc...) = **8**

The parameter value is the additive value assigned to the different complexity requirements you require of the password.

For example, if you wanted to require the generated password to include at least one uppercase letter, at least one lower case letter, and at least one digit, you would set the value at 1 + 2 + 4 = **7**; or:

adclient.random.password.complexity.pattern: 7

The default value for this configuration parameter is 7.

adclient.random.password.length.min

This configuration parameter specifies the minimum character length of a randomly generated Active Directory password.

The default value is 15 characters. For example:

adclient.random.password.length.min: 15

adclient.random.password.length.max

This configuration parameter specifies the maximum character length of a randomly generated Active Directory password.

The default value is 21 characters. For example:

adclient.random.password.length.max: 21

adclient.samba.sync

This configuration parameter specifies whether you want to have the Delinea Agent work in conjunction with Samba. The parameter value can be either true or false. You should set this parameter to false if you do not want any interaction between Delinea and Samba.

If you want the agent to work with Samba, you may need to make changes to your environment or configure additional settings. For Delinea and Samba to operate in the same environment, you need to do the following:

- Check that the samba.base.path configuration parameter specifies the correct path to the Samba binaries.
- Check that the samba.winbind.listen.path configuration parameter specifies the correct path to the Samba winbindd listen path.
- Check that Samba is configured for ADS security.
- Check that Samba belongs to the same REALM as the Delinea Agent.
- Verify that Samba and the Delinea Agent share an Active Directory computer object.
- Set the adclient.samba.sync configuration parameter to true.

For example:

adclient.samba.sync: true
samba.base.path: /usr
samba.windbindd.listen.path: /run/samba/windbindd

For more information about installing and configuring Samba to work with Delinea software, see the *Samba Integration Guide* available on the Delinea web site.

adclient.server.try.max

This configuration parameter specifies the maximum number of servers per domain the agent should attempt to connect to before going into disconnected mode. This parameter is used if the agent is unable to connect to it's primary domain controller to enable it to query DNS for a list of other domain controllers and try each server in the list up to the maximum number of servers you specify. For example, if you have a large number of replica domain controllers for a given domain, you may want to use this parameter to limit the number of servers for the agent to try in order to limit network traffic and improve performance.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer or 0. Setting the parameter value to 0 means that the agent attempts to connect to every server in the list until successful.

The default value is 0.

For example, to allow the agent to attempt to contact up to five domain controllers before going into disconnected mode:

adclient.server.try.max: 5

This parameter is ignored if you have defined a master domain controller for the zone to which the computer belongs because the computer only connects to that domain controller.

> **Note**: This parameter is deprecated for versions of adclient from 4.4.3 to 5.0.*x*. It is available in version 5.1.0 and later.

adclient.skip.inbound.trusts

This configuration parameter specifies whether you want adclient to skip probing inbound trusts for the domaininfomap.

Options are:

- false: If set to false, when building domaininfomap, both two-way and incoming trusts are probed. (Default)
- true: If set to true, when building domaininfomap, only two-way trusts are probed.

Set adclient.skip.inbound.trusts in the centrifydc.conf file. For example:

adclient.skip.inbound.trusts: true

To apply this configuration parameter while adclient is running, follow the recommended sequence:

1. Perform adreload.
2. Rebuild the domaininfomap. Choose a method:
    - Run adflush -t to rebuild the domaininfomap manually.
    - Wait for the next rebuild cycle from adclient.

adclient.skip.unused.outbound.trusts

This configuration parameter specifies whether you want to prevent the agent from sending network queries to outbound trust domains that do not have users in Delinea zones.

If you set this parameter to true, the agent will only send network queries to outbound trust domains that have users in Delinea zones.

If you are manually setting this parameter, the parameter value must be true or false. For example:

adclient.skip.unused.outbound.trusts: true

If the parameter is not explicitly defined in the configuration file or by group policy, its default value is false.

adclient.sntp.enabled

This configuration parameter specifies whether you want to use the Windows Time Service to keep the local system clock in sync with the domain the computer has joined.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be true or false. For example:

adclient.sntp.enabled: true

If the parameter is not explicitly defined in the configuration file or by group policy, its default value is true.

adclient.sntp.poll

This configuration parameter specifies the interval between SNTP clock updates when you are using the Windows Time Service to keep the local system clock in sync with the domain the computer has joined.

In most cases, you set the polling interval using group policy.

If you are manually setting this parameter, the value is the base 2 logarithm of the time in seconds. For example, setting this parameter value to 6 sets the update interval to 64 seconds (26), and a value of 15 sets the update interval to 32768 seconds, or 9.1 hours. For example, to set the update interval to 256 seconds:

adclient.sntp.poll: 8

If the parameter is not explicitly defined in the configuration file or by group policy, its default value is 15 (9.1 hours).

dclient.tcp.connect.timeout

This parameter specifies the timeout of all TCP port probing used in adclient. This parameter default is ten seconds.

adclient.udp.timeout

This configuration parameter specifies the maximum number of seconds to allow to complete UDP binding. The agent will attempt to bind twice. If the first bind request is not complete within the period specified by this parameter, the agent sends a second request with a timeout period that is double the setting of this parameter. If both bind requests fail to complete within the allotted time, the agent sets its status to disconnected.

For example, if you set this parameter to 10 seconds and the bind request is not complete within 10 seconds, the agent sends a second bind request and waits a maximum of 20 seconds for the bind to complete before assuming the computer is disconnected from the network or Active Directory is unavailable.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be a positive integer. The default value for this parameter is 15 seconds. For example:

adclient.udp.timeout: 15

adclient.update.os.interval

This configuration parameter specifies the number of seconds to wait before updating operating system information after adclient starts in disconnected mode.

If you are manually setting this parameter, the parameter value should be a positive integer. The default value for this parameter is 30 seconds. For example:

adclient.update.os.interval: 30

adclient.use.all.cpus

This configuration parameter specifies whether to use all processors on a multi-processor system. The parameter value can be true or false. Setting this parameter to true allows the adclient process to use additional CPUs on a computer to process background tasks in parallel when logging on and can significantly decrease the startup time in sites with a large number of domain controllers.

For example:

adclient.use.all.cpus: true

If the parameter is not explicitly defined in the configuration file, its default value is true. If you change this parameter, you must restart the adclient process for the change take effect.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

adclient.use.tokengroups

This configuration parameter specifies whether the agent should attempt to use the Active Directory tokenGroups attribute on the user object to determine a user's group membership when the Kerberos Privilege Attribute Certificate (PAC) is not available.

In most cases, allowing the agent to use this attribute when necessary is desirable and the default setting for this attribute is true. For example:

adclient.use.tokengroups: true

In mixed-mode domains with both Windows 2000 and Windows 2003 computers, however, the tokenGroups attribute can include Universal groups in the user's group membership list. If you have Universal groups in mixed-mode domains and want to prevent those Universal groups from being included in the user's group membership list, you can set this parameter value to false. Setting this value to false will force the agent to use a slower mechanism for finding group membership instead of the tokenGroups attribute and can result in a slower user login experience, but the results will be consistent with what would be retrieved using the Kerberos PAC.

adclient.user.computers

This configuration parameter specifies whether to allow computer principals to be treated as users with login capabilities when added to the zone. The parameter value can be true or false. The configuration parameter must be set to true to allow Distributed File System support for Samba. Setting this to true may impact performance, however, in domains with heavily-loaded domain controllers or large user and computer populations.

For example:

adclient.user.computers: true

adclient.user.lookup.cn

This configuration parameter specifies whether you want to allow users to be found by their common name (cn) attribute. The parameter value can be true or false.

By default, you can allow users to login using their UNIX profile name, Active Directory displayName, or Active Directory cn attribute. However, allowing users to log on using these additional attributes can require the agent to perform multiple searches to locate a user account in Active Directory. In environments with domain controllers under heavy load or with large user populations, searching Active Directory multiple times might negatively impact performance.

If you want to prevent the agent from attempting to access to user information by the common name, you can set this configuration parameter to false. For example:

adclient.user.lookup.cn: false

The default value for Mac OS X computers is false.

The default parameter value for all other platforms is true.

adclient.user.lookup.display

This configuration parameter specifies whether you want to allow users to be found by their display name (displayName) attribute. The parameter value can be true or false.

By default, you can allow users to login using their UNIX profile name, Active Directory displayName, or Active Directory cn attribute. However, allowing users to log on using these additional attributes can require the agent to perform multiple searches to locate a user account in Active Directory. In environments with domain controllers under heavy load or with large user populations, searching Active Directory multiple times might negatively impact performance.

If you want to prevent the agent from attempting to access to user information by the displayName attribute, you can set this configuration parameter to false. For example:

adclient.user.lookup.display: false

The default value for Mac OS X computers is false.

The default parameter value for all other platforms is true.

adclient.user.name.max.exceed.disallow

When this parameter is set to false, users with a login name longer than eight characters are permitted to log in. When set to true, users with a login name longer than eight characters are not permitted to log in. This configuration parameter applies to local account management in that a local user with rights in addition to the platform-specified limit is not be added to the system.

adclient.version2.compatible

This configuration parameter is used to maintain compatibility with zones created using version 2.0 or 3.0 of Access Manager. The default is true for zones created using the 2.0 or 3.0 console. The default is false for zones created with the 4.x or later console.

If you do not have users or groups that were given access to UNIX computers using an older console, having this parameter set to false results in a performance improvement on Windows 2000 domain controllers. Setting the value to true decreases login performance on Windows 2000 domain controllers.

For example:

adclient.version2.compatible: false

If you have users or groups that were given access to UNIX computers using an older console, you may want to upgrade those users and groups to take advantage of the performance improvements.

To determine whether you have zones and users from an older version of Delinea software , open the console and click **Analyze**. You can then review the Analysis Results and attempt to update user properties, if needed.

adclient.watch.cpu.utilization.info.threshold

When adclient's CPU usage is higher than the value specified by this parameter, cdcwatch will write a message to the INFO log.

The default value is -1, which means that no threshold is set.

If the adclient.use.all.cpus parameter is set to true, the CPU utilization will be divided by the total number of CPUs.

For example, the setting below sets the info CPU utilization threshold to 10%:

adclient.watch.cpu.utilization.info.threshold: 10

adclient.watch.cpu.utilization.warning.threshold

When adclient's CPU usage is higher than the value specified by this parameter, cdcwatch will write a message to the WARN log.

The default value is -1, which means that no threshold is set.

If the adclient.use.all.cpus parameter is set to true, the CPU utilization will be divided by the total number of CPUs.

For example, the setting below sets the warn CPU utilization threshold to 20%:

adclient.watch.cpu.utilization.warn.threshold: 20

adclient.watch.slow.lookup.info.threshold

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request. When a NSS request exceeds this threshold, the NSS module writes information to a message to the INFO log file.

By default, this parameter is set to -1, which means that there is no threshold.

For example, the example below specifies that information goes into the INFO log if the request exceeds 20 milliseconds:

```
adclient.watch.slow.lookup.info.threshold:20
```

adclient.watch.slow.lookup.info.threshold.group

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request categorized by group. When a NSS request exceeds this threshold for the user category, the NSS module writes to a message to the INFO log file.

The group category indicates the following NSS calls: getgrnam* getgrgid*

The nss.watch.slow.lookup.info.threshold.group setting overrides that set by the adclient.watch.slow.lookup.info.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: adclient.watch.slow.lookup.info.threshold.group:35.

adclient.watch.slow.lookup.info.threshold.user

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request categorized by user. When a NSS request exceeds this threshold for the user category, the NSS module writes to a message to the INFO log file.

The user category indicates the following NSS calls: getpwnam* getpwuid* getgrouplist

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the adclient.watch.slow.lookup.info.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: adclient.watch.slow.lookup.info.threshold.user:15.

adclient.watch.slow.lookup.warn.threshold

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

By default, this parameter is set to -1, which means that there is no threshold.

For example: adclient.watch.slow.lookup.warn.threshold:20.

adclient.watch.slow.lookup.warn.threshold.group

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request for the group category. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

The group category indicates the following NSS calls: getgrnam* getgrgid*

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the adclient.watch.slow.lookup.warn.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: adclient.watch.slow.lookup.warn.threshold.group:30.

adclient.watch.slow.lookup.warn.threshold.user

This configuration parameter specifies the adclient threshold (in milliseconds) for the time spent on a complete NSS request for the user category. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

The user category indicates the following NSS calls: getpwnam* getpwuid* getgrouplist

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the adclient.watch.slow.lookup.warn.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: adclient.watch.slow.lookup.warn.threshold.user:25.

adclient.zone.group.count

This configuration parameter provides a calculated value that controls the method used to determine group membership for users. If the calculated value for this parameter is larger than the number of groups a user is a member of, the Delinea Agent iterates over the user's group list to determine group membership. For example, if there are more group profiles defined for the zone than the number of groups the user is a member of, the agent uses the user's group list to determine group membership.

If the calculated value for this parameter is smaller than the typical number of groups a user is a member of, the agent iterates over all of the group profiles enabled for the zone to determine group membership. For example, if there are fewer group profiles defined for the zone than the number of groups the user is a member of, the agent uses the zone's group profile list to determine group membership.

Switching between the two methods for determining group membership may improve the log-in time for some users. You can use this configuration parameter to override the calculated value. For example, if you always want to use the user's group membership list rather than iterate through the list of group profiles defined for the zone, you can set this parameter to an artificially high value. If you always want to use the zone's group profile list rather iterate through the user's group membership list, you can set this parameter to an artificially low value.

For example:

adclient.zone.group.count: 6

addns.tcp.timeout

This configuration parameter controls the amount of time, in seconds, that the addns process waits for responses to its requests for updates.

The parameter value can be any positive integer. The default value of this parameter is 7 seconds:

addns.tcp.timeout: 7

addns.wait.time

This configuration parameter controls the amount of time, in seconds (default 60), that the addns process should wait for another addns process to exit before proceeding.

Because the addns process enables dynamic updates to DNS records on Active Directory-based DNS servers, it includes a mechanism to prevent two addns processes from running at the same time. This configuration parameter value controls how long a addns command request will wait for another addns process to complete its execution before proceeding.

The parameter value can be any positive integer. For example, to set the wait time to 45 seconds:

addns.wait.time: 45

adjust.offset

This configuration parameter specifies the time difference between the local host and the domain that should trigger an adjustment to the local computer's time-of-day setting.

The default parameter value is 5 minutes. With this setting, if the time difference between the local host and the domain controller is less than 5 minutes, the adclient process calls the adjtime function to update the local host time to match the Active Directory domain. If the offset between the local computer and the domain controller is more than 5 minutes, adclient process calls the settimeofday function to update local computer's time.

The parameter value can be any positive integer. For example:

adjust.offset.time: 5

audittrail.audited.command.with.args

This configuration parameter controls whether audit trails for audited command include command parameters. If set to true, the command name and parameters are displayed in the audit trail. If set to false, just the command name is displayed in the audit trail.

The default value is false.

audittrail.Centrify_Suite.Trusted_Path.machinecred.skipda

This configuration parameter specifies whether trusted path audit trail events are sent to the audit installation database in situations where the user is using a computer credential. The default value is true (that is, events are not sent to the audit database). For example:

audittrail.Centrify_Suite.Trusted_Path.machinecred.skipda: true

Events are sent to the system log even if this parameter is set to true.

audittrail.targets

This configuration parameter specifies the target for audit trail information. Possible settings are:

1. Audit information is not sent.
2. Audit information is sent to DirectAudit. This capability is supported by DirectAudit version 3.2 and later.
3. Audit information is sent to the local logging facility (syslog on UNIX systems, Windows event log on Windows systems).
4. Audit information is sent to both DirectAudit and the local logging facility.

If DirectAudit 3.2 or later is installed, the default value is 3 (local logging facility and DirectAudit). Otherwise, the default value is 2 (local logging facility only). For example:

audittrail.targets: 3

In most cases, you set this configuration parameter using group policy.

audittrail.

This parameter specifies whether to override the global audit trail targets. If this parameter is set, the system uses the targets value in the current component; otherwise, the system uses the global configured value.

There are two target settings that can be overridden:

- Whether the system sends the audit trail information to DirectAudit or not
- Whether the system sends the audit trail information to the local logging system or not. On UNIX systems, the local logging system is syslog and on Windows systems it's the Windows event log.

For this setting, you specify a single numeric value to represent where the system will send the audit trail information. (Setting one value to signify two settings is called a bit mask.) The possible settings are as follows:

0	No	No	There is no override to the audit trail target of the current component. The system uses the global audit trail target value.
1	Yes	No	The system overrides just the audit trail target for DirectAudit. This capability is supported by DirectAuditversion 3.2 and later.
2	No	Yes	The system overrides just the audit trail target for the local logging system. If you're using a DirectAuditversion prior to version 3.2, this is the default setting.
3	Yes	Yes	The system overrides both the audit trail targets for DirectAuditand the local logging system. If you're using DirectAuditversion 3.2 or later, this is the default setting.

In most cases, you set this configuration by way of group policy.

audittrail.

This parameter specifies how to calculate where the system sends the audit trail information for a particular component if you have also set the corresponding audittrail.<product>.<component>.overrides parameter.

There are two kinds of audit trail targets that can be overridden:

- Whether to enable the DirectAudit audit trail target for the component or not
- Whether to enable the local logging system audit trail target or not. On UNIX systems, the local logging system is syslog and on Windows systems it's the Windows event log.

For this setting, you specify a single numeric value to represent which audit trail targets are enabled for the component. The possible settings are as follows:

-1	No	No	Use the global audit trail target value. This is the default setting.
0	No	No	Neither the DirectAudit nor the local logging target are enabled for the component.
1	Yes	No	Enable only the DirectAudit audit trail target for the component. This capability is supported by DirectAudit version 3.2 and later.
2	No	Yes	Enable only the local logging audit trail target for the component.
3	Yes	Yes	Enable the audit trail targets for both DirectAudit and the local logging system.

In most cases, you set this configuration parameter using the **Computer Configuration > Policies > Administrative Templates Policy definitions (ADMX files) > Delinea Audit Trail Settings** group policy.

The system calculates the final audit trail targets for a component based on the following information:

- If audittrail.<product>.<component>.overrides is not specified, the system uses the global audit trail target value
- If audittrail.<product>.<component>.overrides is specified, for each target (DirectAudit and local logging), whether the audit trail information will be sent to this target is determined by the following:
    - If audittrail.<product>.<component>.targets is set to -1, or the setting is not overridden in audittrail.<product>.<component>.overrides, the system uses the global audit trail target value
    - If the target is overridden by audittrail.<product>.<component>.overrides and enabled by audittrail.<product>.<component>.targets, the system sends the audit trail information to this target

capi.cache.enabled

This configuration parameter specifies whether the in-process memory CAPI cache is enabled. If the cache is enabled, lookups are sent to the cache before being sent to adclient.

- If the object is found in the cache and has a valid TTL (as configured in the capi.cache.negative.ttl and capi.cache.ttl parameters), the object is returned.
- If the TTL has expired, the lookup is sent to adclient.
- If the object is not found in the cache, the lookup is sent to adclient.

If the object is found in adclient, the cache entry (that is, the key-value and acquisition time stamp) is updated.

If you set this parameter to true, the CAPI cache is enabled.

The following attributes are supported:

- Sid
- _UnixName
- sAMAccountName
- userPrincipalName
- Guid
- Unixid

The default value of this parameter is false. For example:

capi.cache.enabled: false

capi.cache.hash.table.size

This configuration parameter specifies the number of hash map buckets that are allocated if the in-memory CAPI SID cache is enabled through the capi.cache.enabled parameter.

The default value of this parameter is 769. For example:

capi.cache.hash.table.size: 769

capi.cache.log.interval

This configuration parameter specifies the number of seconds between log events that dump information about the performance of the in-memory CAPI SID cache. This parameter takes effect only if the in-memory CAPI SID cache is enabled through the capi.cache.enabled parameter.

- Summary information such as hits, misses, and so on are DEBUG level events.
- Details about the bucket distributions are TRACE level events.

Setting this parameter to 0 disables all hash map log dumps pertaining to the in-memory CAPI SID cache.

The default value of this parameter is 0. For example:

capi.cache.log.interval: 0

capi.cache.max.objects

This configuration parameter specifies the maximum number of objects that are kept in the in-memory CAPI SID cache if the cache is enabled through the capi.cache.enabled parameter. If the number is exceeded, cached objects that are the oldest are replaced with new objects.

The default value of this parameter is ten thousand objects. For example:

capi.cache.max.objects: 10000

capi.cache.negative.ttl

This configuration parameter specifies the number of seconds that a negative cached SID object remains in the in-memory CAPI SID cache before it is refreshed. This parameter takes effect only if the in-memory CAPI SID cache is enabled through the capi.cache.enabled parameter.

The default value of this parameter is 3,600 seconds. For example:

capi.cache.negative.ttl: 3600

capi.cache.ttl

This configuration parameter specifies the number of seconds that a positive cached SID object remains in the in-memory CAPI SID cache before it is refreshed. This parameter takes effect only if the in-memory CAPI SID cache is enabled through the capi.cache.enabled parameter.

The default value of this parameter is 3,600 seconds. For example:

capi.cache.ttl: 3600

db2.implement.pam.ignore.users

Starting with Delinea DB2 agent 5.2.3, this configuration parameter specifies whether the Delinea DB2 agent checks pam.ignore.users for a list of users to authenticate locally, without first attempting to authenticate those users in Active Directory.

By default, the Delinea DB2 agent authenticates users in Active Directory first. If users do not exist in Active Directory, the Delinea DB2 agent then authenticates users locally.

If you set this parameter to true, users defined in the pam.ignore.users list are authenticated locally only (that is, no attempt is made to authenticate them in Active Directory first). For example:

db2.implement.pam.ignore.users: true

To specify that an Active Directory authentication attempt should be made for all users, and that local authentication be attempted only for users not in Active Directory, set this parameter to false:

db2.implement.pam.ignore.users: false

If you change the setting of this parameter, restart the DB2 instance to activate the new setting.

db2.user.zone_enabled

This configuration parameter specifies whether to constrain the DB2 agent authentication to zone enabled Active Directory users only. By default, the DB2 agent authenticates all Active Directory users even if the Active Directory user is not in the zone. To constrain the authentication to zone enabled Active Directory users only, add the following parameter to the /etc/centrifydc/centrifydc.conf file:

db2.user.zone_enabled.db2_instance_name: true

In this parameter, db2_instance_name is the name of the DB2 instance (for example, db2inst1).

After you add this parameter, restart the DB2 instance to pick up the new setting.

db2.userpass.username.lower

This configuration parameter specifies whether the DB2 userpass plugin is used to convert the user name to lowercase before attempting authentication (true) or not make the conversion (false, the default).

dc.dead.cache.refresh

This configuration parameter specifies how long, in seconds, to keep in cache the fact that a domain controller is dead.

The default value is 60 seconds. For example:

dc.dead.cache.refresh: 60

dc.live.cache.refresh

This configuration parameter specifies how long, in seconds, to keep in cache the fact that a domain controller is alive.

The default value is 3600 seconds (one hour). For example:

dc.live.cache.refresh: 3600

dc.penalty.time

This configuration parameter controls how long a domain controller that has failed is considered less preferable to the other domain controllers in the forest that either have not failed or have failed farther back in time.

The default setting is 3600 seconds (one hour).

This parameter helps you avoid domain controllers that appear to be alive, but when they are selected exhibit higher level failures such as crashed, tombstoned, or dead netlogon service.

The value specifies the number of seconds. For example, the following specifies two hours:

dc.penalty.time: 7200

dns.alive.resweep.interval

This configuration parameter specifies the amount of time to wait, when DNS is active, before triggering a DNS server sweep to see if any DNS servers are responding faster than the current one.

The adclient process periodically checks in the background to see if any DNS servers are available with faster response times than the currently active DNS server. This parameter, dns.alive.resweep.interval, determines how often this check, or sweep, occurs. The default is one hour (3600 seconds).

For the sweep, the dns.sweep.pattern parameter determines the probe pattern that is used to find a live DNS server; that is, it sets:

- The protocol to use (TCP or UDP)
- The amount of time to wait for a response.

The DNS server that responds fastest is selected, is cached in memory, and is used for all DNS requests until one of the following occurs:

- It stops responding.
- A new server sweep discovers a faster DNS server and replaces it.
- Adclient is stopped and restarted.

If the newly selected server is different than the previous server, the kset.dns.server file is updated with the address of the newly selected server.

The default value for this parameter is 3600 seconds.

The parameter value must be a positive integer. For example:

dns.alive.resweep.interval: 3600

dns.block

This configuration parameter specifies the list of domain controllers that should be filtered out when resolving the domain controller to contact through DNS. This configuration parameter enables you to prevent the adclient process from attempting to contact domain controllers that are known to be inaccessible, for example, because they reside behind a firewall, or domain controllers that shouldn't be contacted, for example, because of their physical location or because they are no longer valid domain controllers for the site.

The parameter value can be one or more fully-qualified domain controller server names. If you are specifying more than one domain controller name, the names can be separated by commas or spaces. For example:

dns.block: ginger.ajax.org,salt.ajax.org,nc1.sea.ajax.org

In most cases, you set this configuration parameter using group policy.

If you don't specify a value for this parameter, access is not blocked for any domain controllers.

Delinea

dns.cache.negative

This configuration parameter specifies whether to cache negative DNS responses. A negative response is returned when a DNS server is not found. By storing a negative result in the cache, the agent does not look for a server that was previously not found.

Set this parameter to true to cache negative DNS responses or false to not cache negative responses. When this parameter is false, the system attempts to respond to all requests. A cached response expires after the amount of time specified by the dns.cache.timeout parameter (default value is 300 seconds).

The default is true; for example:

dns.cache.negative:true

dns.cache.timeout

This configuration parameter specifies the amount of time, in seconds, before a cached DNS response expires.

The default value is 300 seconds.

Specify a positive integer; for example:

dns.cache.timeout:300

dns.dc.domain_name

This configuration parameter can be used to specify the domain controller host names if your DNS is not configured to use Active Directory. In most cases, you should not use this configuration parameter in a production environment because Active Directory automatically updates DNS with fail-over and replica servers optimized for the Active Directory site configuration. This configuration parameter is used primarily for configuring an evaluation environment when the DNS server is on a UNIX computer and can't provide the _ldap service records.

To set this parameter, the Active Directory domain name must be specified as the last portion of the configuration parameter name, and the parameter value is the host name of the domain controller. For example, if the Active Directory domain is acme.com and the domain controller for that domain is coyote.acme.com:

dns.dc.acme.com: coyote.acme.com

> **Note**: You must specify the name of the domain controller, not its IP address. In addition, the domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify multiple servers for a domain, use a space to separate the domain controller server names. For example:

dns.dc.lab.test: dc1.lab.test dc2.lab.test

dns.dead.resweep.interval

This configuration parameter specifies the amount of time to wait, in seconds, when DNS is down, before triggering a DNS server sweep to see if any DNS servers are alive.

If the current DNS server times out on a request (does not respond within the interval and number of retries specified by dns.tcp.timeout or dns.udp.timeouts), the agent attempts to acquire another DNS server. If it fails to find a live server, DNS is considered down and the agent waits for the interval specified by this parameter, dns.dead.resweep.interval, before attempting to acquire another DNS server.

The default is 60 seconds.

The parameter value must be a positive integer. For example:

dns.dead.resweep.interval: 60

dns.gc.domain_name

This configuration parameter can be used to specify the domain controller used as the global catalog if your DNS is not configured to use Active Directory. In most cases, you do not use this configuration parameter in a production environment. This configuration parameter is used primarily for configuring an evaluation environment when the DNS server is on a UNIX computer and can't provide the _gc service records.

To set this parameter, the Active Directory domain name must be specified as the last portion of the configuration parameter name, and the parameter value is the host name of the domain controller. For example, if the Active Directory domain is arcade.com and the domain controller for that domain is fire.arcade.com:

dns.gc.arcade.com: fire.arcade.com

> **Note**: You must specify the name of the domain controller, not its IP address. In addition, the domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify multiple servers for a domain, use a space to separate the domain controller server names. For example:

dns.dc.lab.test: dc1.lab.test dc2.lab.test

dns.query.all.servers

This configuration parameter specifies whether the DNS subsystem should try all live DNS servers until either the lookup succeeds or the list is exhausted.

When this parameter is set to true (the default), DNS tries each server on the list of all DNS servers in /etc/resolv.conf (or dns.servers) one-by-one until either the list is exhausted or the object is resolved. By default, this configuration parameter is configured as true:

dns.query.all.servers: true

When this parameter is set to false, the DNS subsystem stops querying after the first "record not found" response.

This feature is useful in environments that contain multiple DNS servers that do not all hold the same records (and are therefore not all aware of the same AD domains).

dns.servers

This configuration parameter specifies a space separated list of IP addresses of DNS servers that are used to resolve domain controller names. Set this parameter if a computer running Mac OS X 10.7 or later cannot connect to a domain controller through a VPN connection.

Starting with Mac OS X 10.7, /etc/resolv.conf is no longer used for domain controller name resolution. Therefore, some VPN programs no longer update DNS server information in /etc/resolv.conf when signing on. On computers running Mac OS X 10.7 and later, this can result in the computer not being able to connect to a domain controller through a VPN if the DNS server locations are not specified as described here.

The following example shows the setting of two IP addresses for DNS servers:

dns.servers: 111.22.333.4 555.66.777.8

dns.sort

This configuration parameter determines whether to sort by speed during background sweeps or to pick the first DNS server that responds.

> **Note**: This parameter only applies to *background* sweeps. During *initial* sweeps, the first server to respond is always chosen, regardless of how dns.sort is set.

Generally, the first server in the list (as specified in /etc/resolv.conf or by the dns.servers parameter) responds first. However, if a server was previously chosen, and is still configured in /var/centrify/kset.dns.server, it is always tried first regardless of how dns.sort is set.

This parameter is useful if you have multiple DNS servers specified in /etc/resolv.conf, some of which are not compatible with DirectControl. If you list the DirectControl-compatible first, and set this parameter to false, an incompatible server will never be chosen unless the compatible servers are unavailable.

Set the value of this parameter to true to sort by speed. Set the value to false to select the first server that responds.

The default is to sort by speed (true); for example:

dns.sort: true

dns.sweep.pattern

This configuration parameter specifies a comma separated list to use when scanning for live DNS servers. For each item in the list, specify the type of scan (t for TCP; u for UDP) and the number of seconds to wait for a response.

For example, the following pattern:

dns.sweep.pattern: t1, u1,u2

specifies:

- A TCP scan with a one second wait for a response
- A UDP scan with a one second wait for a response
- Another UDP scan with a two second wait for a response

For each value, all known DNS servers are queried. If the kset.dns.server file exists, the server it defines is queried first.

For initial DNS server acquisition, the first DNS server to respond is chosen, at which point the sweep is terminated. Since the kset.dns.server file is queried first, the server it defines is most likely to be selected. Otherwise, the first server specified in /etc/resolv.conf responds first

For background DNS sweeps, the entire sweep pattern is completed, at which point the fastest server to respond is chosen and the sweep is terminated.

If a new DNS server is selected, the kset.dns.server file is updated with its address.

If the end of the list is reached and no DNS servers respond, DNS is considered down. A new sweep begins after the period of time specified by the dns.dead.resweep.interval.

The default pattern for Linux and Unix is:

dns.sweep.pattern: t1,u1,u1,t2,u2,u2

The default pattern for OS X is:

dns.sweep.pattern: u1,u1

dns.tcp.timeout

This configuration parameter specifies the amount of time, in seconds, to wait before re-sending a TCP request, when there is no response from the current DNS server. If the current server does not respond to this request, it is considered down, which triggers a sweep to acquire a new server as specified by the dns.sweep.pattern parameter. The new server becomes the selected server (it is cached in memory and its address is put in kset.dns.server), and it attempts to handle the DNS request.

The default value is 1 second. You may specify only one TCP retry.

Specify a positive integer; for example:

dns.tcp.timeout: 1

This parameter specifies the timeout values for TCP requests. Use dns.udp.timeouts to specify timeout values for UDP requests.

dns.udp.timeouts

This configuration parameter specifies the number of times to re-send a UDP request, and the number of seconds to wait for each, when there is no response from the current DNS server to a UDP request. Specify a comma separated list of values, up to three entries. If the current server does not respond to any of the requests, it is considered down, which triggers a sweep to acquire a new server as specified by the dns.sweep.pattern parameter. The new server becomes the selected server (it is cached in memory and its address is put in kset.dns.server), and it attempts to handle the DNS request.

The default value on Linux and Unix is three retries of 1, 2, and 4 seconds, respectively.

The default value on OS X is 1 second.

Specify a positive integer; for example:

dns.udp.timeouts: 1, 2, 4

This parameter specifies the timeout values for UDP requests. Use dns.tcp.timeout to specify timeout values for TCP requests.

domain.dead.cache.refresh

This configuration parameter specifies how long, in seconds, to keep in cache the fact that a domain is dead (that is, the domain does not contain any live domain controllers).

The default value is 60 seconds. For example:

domain.dead.cache.refresh: 60

domain.live.cache.refresh

This configuration parameter specifies how long, in seconds, to keep in cache the fact that a domain is alive (that is, the domain contains at least one live domain controller).

The default value is 3600 seconds (one hour). For example:

domain.live.cache.refresh: 3600

fips.mode.enable

This configuration parameter indicates whether FIPS 140-2 compliant algorithms are used in the authentication protocols. FIPS 140-2 compliance is available for authentication using Kerberos and NTLM with the following caveats and requirements:

- FIPS mode is available on Centrify Agents version 5.0.2 or later but only on specific UNIX platforms. See the NIST validation entry for the Centrify FIPS mode for the current list of supported platforms.
- Domain controllers must be at Windows 2008 domain functional level or greater. If the domain controller domain functional level does not meet the required level, adclient does not start and returns an error message.
- FIPS 140-2 compliance uses only the following algorithms: AES128-CTS or AES256-CTS encryption types, RSA for public key generation, DSA for digital signature generation and SHA1, SHA256, SHA384 or SHA512 for hashing.
- Inter-realm keys for the AES128-CTS or AES256-CTS encryption types must be established between any trusted domains to enable Active Directory users to log on to a joined computer (see the ksetup utility to set up inter-realm keys).
- FIPS mode only allows NTLM pass-through authentication over SChannel; FIPS mode is not available for 'NTLM authentication over SMB or SMB2.

In most cases, you set this configuration parameter using group policy. As long as the UNIX computer is running a supported platform, this policy sets the fips.mode.enable configuration parameter to true and restarts adclient.

> **Note**: The administrator must explicitly add the centrifydc_fips.xml or centrifydc_fips.adm group policy template on the domain controller to set fips.mode.enable. The template needs to be imported to just one domain controller in a forest.

If you are manually setting this parameter, the parameter value must be true or false. For example, to enable FIPS 140-2 compliant algorithms, set the following:

fips.mode.enable: true

The default is false.

After manually setting this parameter, you must restart adclient to enable FIPS mode.

There are several restrictions and rules governing the use of FIPS mode. For example:

- Prevalidated groups and users that use FIPS mode to log in when disconnected must have their Active Directory msDS-SupportedEncryptionTypes attribute set to at least 0x18 (prevalidated login for users in FIPS mode requires Kerberos AES 128- or 256-bit encryption).See adclient.prevalidate.allow.groups and adclient.prevalidate.allow.users for the full explanation of the Active Directory msDS-SupportedEncryptionTypes options.
- The value of the corresponding Windows policy (**Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Option > System Cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing**) has no effect on the Windows, Linux, UNIX, or Mac OS X computers managed through the Delinea Agent. You must use the configuration parameter or the Delinea policy to enable FIPS mode.

The following configuration parameters affect adclient operation when FIPS mode is enabled:

- adclient.krb5.keytab.clean.nonfips.enctypes: Set this configuration parameter to true to have adclient scan the computer's keytab file and remove all non-AES encryption keys for service principal names (SPNs) during startup. (The default is false.)
- adclient.krb5.permitted.encryption.types: If you include the arcfour-hmac-md5 encryption type in this configuration parameter AND adclient.krb5.extra_addresses is true, adclient generates the MD4 hash for the computer password and saves it in the keytab file.

For more information about using FIPS encryption, see the *Administrator's Guide for Linux and UNIX*.

log

This configuration parameter defines the level of detail written to the agent log file. The log level works as a filter to define the type of information you are interested in and ensure that only the messages that meet the criteria are written to the log. For example, if you want to see warning and error messages but not informational messages, you can change the log level from INFO to WARN.

The parameter value can be FATAL, ERROR, WARN, INFO, DEBUG, or TRACE. For example:

log: WARN

You can also modify this configuration parameter to define a different logging level for specific library messages. For example:

log: info
log.pam: debug

logger.facility.adclient

This configuration parameter defines the syslog facility to use for logging general adclient activity. You can specify separate syslog facilities for logging general adclient messages, adclient auditing messages, and adnisd messages. This parameter's value can be any valid syslog facility. For example, you can set this parameter to log messages to auth, authpriv, daemon, security, or localn facilities.

The default facility is auth. For example:

logger.facility.adclient: auth

> **Note**: You can specify other process names for logging, or use an asterisk (*) to specify the default facility to use for all agent processes. For example, you can specify logger.facility.*: auth in the configuration file to direct all agent processes send messages to the auth facility of syslog.

logger.facility.adclient.audit

This configuration parameter defines the syslog facility to use for logging adclient auditing messages. You can specify separate syslog facilities for logging general adclient messages, adclient auditing messages, and adnisd messages. This parameter's value can be any valid syslog facility. For example, you can set this parameter to log messages to auth, authpriv, daemon, security, or localn facilities.

The default facility is auth. For example:

logger.facility.adclient.audit: auth

If this parameter is not defined in the configuration file, the audit messages are logged in the facility defined for the logger.facility.adclient parameter.

logger.facility.diag

This configuration parameter defines the syslog facility to use for logging diagnostic messages. Diagnostic messages are intended to help you troubleshoot operations and trace all of the LDAP, Kerberos, NTLM and RPC messages that are generated for the following tasks:

- adjoin operations
- adleave operations
- lookup object operations
- authentication operations
- log on operations
- password change

This parameter enables you to specify a separate syslog facilities for logging diagnostic from the facility used to log general adclient messages, adclient auditing messages, and adnisd messages. This parameter's value can be any valid syslog facility. For example, you can set this parameter to log messages to auth, authpriv, daemon, security, or localn facilities.

The default facility is auth. For example:

logger.facility.diag: auth

You should note that diagnostic messages are only logged if you enable logging with the addebug command. If the parameter is not defined in the configuration file, the messages are logged in the default facility or the facility defined for the logger.facility.adclient parameter.

logger.memory.bufsize

This configuration parameter specifies the default size for the in-memory circular log buffer. The in-memory circular log buffer is only enabled if the adclient watchdog process is forced to restart the adclient process. The default parameter value is 128K. You should not manually set this parameter value in the configuration file unless you are instructed to make the setting by Centrify Support.

logger.memory.enabled

This configuration parameter specifies whether the in-memory circular log buffer is enabled. The in-memory log buffer should only be enabled automatically if the adclient watchdog process is forced to restart the adclient process. Therefore, the default value for this parameter is false. You should not manually set this parameter value in the configuration file unless you are instructed to make the setting by Delinea Support.

logger.memory.log

This configuration parameter specifies the default log level for the in-memory circular log buffer. The in-memory circular log buffer is only enabled if the adclient watchdog process is forced to restart the adclient process. The default value for this parameter is DEBUG. You should not manually set this parameter value in the configuration file unless you are instructed to make the setting by Delinea Support.

logger.queue.size

This configuration parameter controls the maximum number of messages that may be queued before they are sent to syslog. The messages in the queue are sent to syslog asynchronously. During normal operation, if the number of messages in the queue reaches the value set for this parameter, no new messages are added until the number of messages in the queue decreases below the maximum number you have specified.

Each message consumes about 100 bytes of storage in the message queue.

If the logging level is set to DEBUG, this parameter's value is automatically multiplied by a factor of 4 to allow additional messages to be logged.

The parameter value must be a positive integer. For example:

log.queue.size: 256

> **Note**: Setting this parameter to zero (0) disables the message queue, and causes all log messages to be written to the syslog facility synchronously. In most cases, disabling the message queue degrades system performance, and in extreme cases, may cause a dead lock with the syslog daemon during log rotations. Therefore, Delinea recommends that you never set this parameter value to 0.

If this parameter is not defined in the configuration file, its default value is 256 KB.

If you change this parameter, you must restart the agent, adclient, for the change take effect.

lrpc.connect.timeout

This configuration parameter specifies the number of seconds the NSS or PAM service should wait for a response from the agent during an initial connection attempt. If the initial connection to adclient takes longer than specified by this parameter, the service will time out and terminate the attempt to connect. In most cases, there's no need to modify this parameter.

The parameter value must be a positive integer. For example:

lrpc.connect.timeout: 5

If this parameter is not defined in the configuration file, its default value is 5 seconds.

lrpc.session.timeout

This configuration parameter specifies the maximum number of seconds to keep the adclient connection open to respond to contextdependent requests, such as pwgetent or lsgroup requests. Lowering this value reduces the chance of a multi-threaded program being affected by an adclient restart, but may cause slow context-dependent commands to fail to return results because the session times out before the command completes its operation. Increasing the value of this parameter reduces the overhead of re-establishing a connection for multiple requests.

For example:

ldap.session.timeout: 30

lrpc.timeout

This configuration parameter specifies the number of seconds the local client should wait for a response from the agent before ending a requested operation.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be an integer greater than zero. The following example sets the inactive client timeout to 5 minutes:

lrpc.timeout: 300

If this parameter is not defined in the configuration file, its default value is 5 minutes.

Although in some environments increasing or decreasing the value of this parameter may be beneficial to optimize agent operations and Active Directory for your network topology, you should take care in changing this setting. For example, in most cases, you should not decrease this value because of the potential problems it may cause when transferring data. If you set this value too low and have a slow connection or a large amount of data to be transferred, the local client may end the operation prematurely and prevent the data transfer from completing successfully.

secedit.system.access.lockout.allowofflinelogin

This configuration parameter specifies whether to allow users to log in when the user account is locked out and the computer is not connected to Active Directory. The default value is false (that is, users cannot log in). For example:

secedit.system.access.lockout.allowofflinelogin: false

You can also set this parameter using group policy.

queueable.random.delay.interval

This configuration parameter specifies a delay, in minutes, for activities to stagger background tasks. Once defined, scheduling of those background tasks calculates a random period of time within the interval and adds the same time to the delay of those tasks. If you change the interval setting, however, the period of time is recalculated. The default setting is 0 and no delay.

Customizing Kerberos-Related Configuration Parameters

This chapter describes the configuration parameters that affect the operation of Kerberos-related activity on the local host computer.

adclient.dc.switch.update.krb5.conf

The adclient.dc.switch.update.krb5.conf configuration parameter specifies that adclient updates the krb5.conf file immediately with the current domain controller when adclient switches the domain controller, such as in failover situations. This can be helpful in situations where third party applications use the krb5.conf file to locate the domain controllers that adclient uses.

By default, this parameter is true.

If this parameter is set to false, then adclient does not update the krb5.conf file with the updated domain controller information immediately but at the next update interval that's specified by the krb5.config.update parameter.

adclient.krb5.allow_weak_crypto

This configuration parameter specifies whether to allow weak encryption types for Kerberos authentication. When this parameter is set to false, then weak encryption types (as noted in the Encryption types section of the kdc.conf file) are filtered out of the following lists:

- default_tgs_enctypes
- default_tkt_enctypes
- permitted_enctypes.

The default value for this parameter is false, which may cause authentication failures in existing Kerberos infrastructures that do not support strong crypto. Users in affected environments should set this parameter to true until their infrastructure adopts stronger ciphers.

By default, this parameter is set to false.

adclient.krb5.allow_weak_crypto: false

adclient.krb5.autoedit

This configuration parameter specifies whether the agent should automatically update the Kerberos configuration file with new information, such as domains and IP addresses, as the agent discovers this information.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be true or false. In most cases, this parameter should be set to true to allow the agent to maintain the configuration files automatically. For example:

adclient.krb5.autoedit: true

If this parameter is not defined in the configuration file, its default value is true.

adclient.krb5.cache.renewal.service.accounts

This configuration parameter specifies which service accounts are renewed automatically.

The parameter value can be a comma-separated list of service accounts, or the name of a file that contains the list of service accounts.

For example, if you specify a file that contains the service accounts using the file: keyword and a file location:

adclient.krb5.cache.renewal.service.accounts: file:/etc/centrifydc/service_accts.lst

The default value of this parameter is file:/etc/centrifydc/service_accts.lst as shown in the example.

adclient.krb5.ccache.dir

The adclient.krb5.ccache.dir parameter specifies the directory where Kerberos ccache files are stored when krb5.cache.type is FILE.

This is useful when kerberos applications in docker containers use the kerberos cache files. This parameter, in conjunction with adclient.krb5.ccache.dir.secure.usable.check enables volume bind mapping so that kerberos cache files in the host OS are available to the docker containers.

Default is empty string.

- If adclient.krb5.ccache.dir is not configured or set to default empty string, then:

  The system default ccache directory is used. If a default_ccache_name exists in the [libdefaults] stanza of krb5.conf, it is removed.

- If adclient.krb5.ccache.dir is specified, AND adclient.krb5.ccache.dir.secure.usable.check is false, then:

  The specified directory is used for the default_ccache_name in the [libdefaults] stanza of krb5.conf.

- If adclient.krb5.ccache.dir is specified, AND adclient.krb5.ccache.dir.secure.usable.check is true, BUT the kerberos cache directory is neither secure nor usable, then:

  The system default ccache directory is used. If a default_ccache_name exists in the [libdefaults] stanza of krb5.conf, it is removed.

- If adclient.krb5.ccache.dir is specified, AND adclient.krb5.ccache.dir.secure.usable.check is true, AND the kerberos cache directory is secure and usable then:

  The specified directory is used for the default_ccache_name in the [libdefaults] stanza of krb5.conf.

  > **Note**: When the ccache type is KCM, the klist lists KCM caches and file ccaches under the system default ccache directory. If the ccache directory is changed when ccache type is FILE, the newly created file ccaches might not be listed when ccache type is switched to KCM.

adclient.krb5.ccache.dir.secure.usable.check

The adclient.krb5.ccache.dir.secure.usable.check parameter specifies whether to perform a secure and usability check on a configured Kerberos ccache directory. Only used when adclient.krb5.ccache.dir set. Options are:

- false — Default. No action taken.
- true — If adclient.krb5.ccache.dir is configured, then adclient.krb5.ccache.dir.secure.usable.check checks the specified directory.

For the kerberos cache directory to be secure and usable it must meet the following criteria:

- the directory exists
- the directory is not a symlink
- the directory is root owned
- the directory is world writable and has sticky bit set

adclient.krb5.conf.file.custom

This configuration parameter enables the merging of custom krb5.conf entries into the original krb5.conf file. To use this parameter, you specify the keyword file: and the absolute path to a syntactically valid custom krb5.conf file.

For example:

adclient.krb5.conf.file.custom: file:/etc/custom.conf

By default, this parameter is not enabled, and the default value is an empty string.

After you enable this parameter, when krb5.conf is regenerated the additional directives in the custom krb5.conf file are merged into the original krb5.conf file, and conflicting lines are discarded.

The required format of the custom krb5.conf file is as follows:

```
[libdefaults]
keyword1 = value1
keyword2 = value2
[domain_realm]
domain = realm
hostname = realm
[realms]
REALM1 =
REALM2 =
[appdefaults]
to-be-copied-as-is
[capaths]
to-be-copied-as-is
[dbdefaults]
to-be-copied-as-is
[dbmodules]
to-be-copied-as-is
[kadmin]
to-be-copied-as-is
[kdc]
to-be-copied-as-is
[kdcdefaults]
to-be-copied-as-is
[logging]
to-be-copied-as-is
[login]
to-be-copied-as-is
[otp]
to-be-copied-as-is
[password_quality]
to-be-copied-as-is
[plugins]
to-be-copied-as-is
```

When you use this parameter, the following actions take place when the krb5.conf file is regenerated:

- For the directives [libdefaults], [domain_realm], and [realms], the new keyword = value pairs from the custom krb5.conf file are added to the corresponding directive in the original krb5.conf file.
- New realms from the custom krb5.conf file are added under [realms] in the original krb5.conf file.
- If a keyword already exists in the original krb5.conf file, the keyword entry from the custom file is discarded.
- For the additional sections [appdefaults], [capaths], [dbdefaults], [dbmodules], [kaadmin], [kdc], [kdcdefaults], [logging], [login], [otp], and [plugins], the entire section from the custom file is added directly into the original krb5.conf file, and any existing entries in those sections in the original

krb5.conf file are overwritten.

- Warning messages are displayed in the log for every conflict.

  Note: The specified custom krb5.conf file must be owned by root.

  Note: To use this parameter in a Mac environment, the configuration parameter adclient.krb5.autoedit must be set to true.

adclient.krb5.conf.domain_realm.anysite

This configuration parameter specifies whether or not to search for all domain controllers in a kerberized realm or just the domain controllers within the current, preferred site.

If this parameter is set to true, then the system will list all reachable domain controllers in a kerberized realm, regardless of which site they're located in.

If this parameter is set to false, then only the domain controller in the current, preferred site is listed.

For example:

adclient.krb5.conf.domain_realm.anysite: true

If this parameter is not defined in the configuration file, its default value is false.

adclient.krb5.extra_addresses

This configuration parameter specifies 0, 1, or more IP addresses. The Delinea Agent adds these IP addresses to the host computer's own IP address when it makes a Kerberos authentication request that includes IP addresses. Multiple addresses accommodate authentication in a network that uses NAT.

The IP addresses in this parameter should be in dotted quad form, each address separated from the next by a comma. As an example:

adclient.krb5.extra_addresses: 192.68.21.189,192.68.35.2

adds two IP addresses to the host machine's own IP address.

Note that this configuration parameter sets the Kerberos configuration parameter extra_addresses in krb5.conf.

This parameter has no effect unless adclient.krb5.use.addresses is set to true.

If this parameter is not defined in the configuration file, its default value is empty, which defines no extra IP addresses.

adclient.krb5.keytab.clean.nonfips.enctypes

This configuration parameter specifies whether adclient scans the computer's keytab file and removes any non-AES encryption keys for service principal names during startup. The default is false.

Use this configuration parameter to remove the keys for encryption types that are not supported when you enable FIPS mode (see fips.mode.enable). To remove the non-AES keys, enter the following

adclient.krb5.keytab.clean.nonfips.enctypes: true

> **Note**: If you specify arcfour-hmac-md5 in the adclient.krb5.permitted.encryption.types configuration parameter, the MD4 hash of the computer password is generated and saved in the keytab file.

adclient.krb5.keytab.entries

This configuration parameter specifies the number of entries that the agent maintains in the Kerberos key table for a service principal.

This value determines the number of key versions that are kept per service principal. Its value must be a positive integer. For example:

adclient.krb5.keytab.entries: 3

If this parameter is not defined in the configuration file, its default value is 3 entries.

adclient.krb5.keytab.use.all.etypes

By default, the Delinea Agent for *NIX does not always generate krb5.keytab entries of all supported encryption types. Instead, the agent generates krb5.keytab entries of the types specified in the adclient.krb5.tkt.encryption.types and adclient.krb5.permitted.encryption.types parameters.

The adclient.krb5.keytab.use.all.etypes parameter specifies whether or not to write the krb5.keytab entries of all encryption types, regardless of the adclient.krb5.tkt.encryption.types and adclient.krb5.permitted.encryption.types setting.

By default, the adclient.krb5.keytab.use.all.etypes parameter is set to false.

If you set the adclient.krb5.keytab.use.all.etypes parameter to true, then the agent generates all types of keys in the krb5.keytab file, regardless of the adclient.krb5.tkt.encryption.types and adclient.krb5.permitted.encryption.types setting.

adclient.krb5.password.change.hook

The adclient.krb5.password.change.hook configuration parameter specifies the full path of the command that adclient runs after adclient has changed a password and updated the krb5.keytab file.

By default, this parameter is empty.

Here's an example where you would use this parameter:

You want adclient to maintain an external keytab file for the ftp service that a non-privileged user "ftp" runs. You need adclient to copy only the ftp keys from the machine keytab file to a keytab file that only the "ftp" user can read. You can create a script, for example, /var/ftp/create_keytab_for_ftp.sh to help you to do this:

#/bin/sh

/usr/sbin/adkeytab -o -P ftp -K /var/ftp/ftp.keytab -b /etc/krb5.keytab && \

chown ftp:ftp /var/ftp/ftp.keytab

And then you add the script to the adclient.krb5.password.change.hook parameter:

adclient.krb5.password.change.hook: /var/ftp/create_keytab_for_ftp.sh

adclient.krb5.password.change.interval

This configuration parameter specifies the number of days in the interval between the last Active Directory password change for the computer account and the next password change for the account. At the interval, Active Directory prompts for a new account password. The agent then automatically generates a new password for the computer account and issues the new password to Active Directory.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be an integer equal to or greater than zero. If the value is zero, then the change interval is turned off and the account is not prompted for password change. For example:

adclient.krb5.password.change.interval: 28

If this parameter is not defined in the configuration file, its default value is 28 days.

adclient.krb5.password.change.verify.interval

This configuration parameter controls how long adkeytab waits between attempts to verify password changes. For example, to set the interval between verification attempts to 600 seconds (10 minutes), enter the following:

adclient.krb5.password.change.verify.interval: 600

The default setting for this parameter is 300 seconds (5 minutes).

You can specify the number of password change verifications that adkeytab attempts by using the adclient.krb5.password.change.verify.retries configuration parameter.

adclient.krb5.password.change.verify.retries

This configuration parameter controls how many times adkeytab tries to verify password changes running in the background.

In some Active Directory environments, such as those employing a read-only domain controller (RODC), Kerberos password changes may not be verified through adclient due to a replication delay. As a result of this delay, the new password is not saved to the keytab file. When this parameter is set to a value other than 0, adclient will retry verification of the new password a corresponding number of times.

If your RODC has latency problems, you may want to address this by setting adkeytab to attempt to verify password changes multiple times. For example, to direct adkeytab to attempt a total of 4 password change verifications, you would set this parameter to 3 as follows:

adclient.krb5.password.change.verify.retries: 3

The time between verification attempts can be set using the adclient.krb5.password.change.verify.interval configuration parameter.

The default setting for this parameter is 0, meaning that adkeytab will not try to verify password changes after the initial attempt.

adclient.krb5.passwd_check_s_address

This configuration parameter specifies whether Kerberos should ignore the source address on private messages. This setting is useful when Active Directory uses NAT.

The parameter value can be true or false. The default value for this parameter is true. For example:

adclient.krb5.passwd_check_s_address: false

adclient.krb5.permitted.encryption.types

This configuration parameter specifies the types of encryption that can be used in Kerberos client credentials.

The parameter value must be one or more encryption types, separated by a space. For example:

adclient.krb5.permitted.encryption.types: arcfour-hmac-md5 des-cbc-md5

If this parameter is not defined in the configuration file, the default encryption types permitted are:

- Windows 2000 server and Windows Server 2003: arcfour-hmac-md5, des-cbc-md5, and des-cbc-crc.
- Windows Server 2008 domain functional level supports these additional types:

  aes128-cts and aes256-cts. Although you can specify these types in an environment other than 2008 domain functional level, they are not useful and may cause extra network round trips during the authentication process.

adclient.krb5.permitted.encryption.types.strict

The adclient.krb5.permitted.encryption.types.strict parameter controls whether to add to or replace the encryption types specified in the setting, permitted_enctypes, in krb5.conf with the encryption types specified in the setting, adclient.krb5.permitted.encryption.types, in centrifydc.conf.

- When adclient.krb5.permitted.encryption.types.strict is false (default), then:

  The encryption types listed in adclient.krb5.permitted.encryption.types in centrifydc.conf,are added to the list of encryption types in permitted_enctypes in krb5.conf.

  This only ensures that what is specified in centrifydc.conf is present in krb5.conf. It does not remove unknown items.

- When adclient.krb5.permitted.encryption.types.strict is set to true, then:

  The encryption types listed in adclient.krb5.permitted.encryption.types in centrifydc.conf replace the encryption types specified in the setting, permitted_enctypes, in krb5.conf.

  The permitted encryption types in krb5.conf exactly match the permitted encryption types in centrifydc.conf. Extra or unknown encryption types are removed.

Example:

adclient.krb5.permitted.encryption.types.strict: false

- false — Default is false. No change in behavior. permitted_enctypes are updated from the centrifydc.conf file.

  Items from centrifydc.conf are added, if they were not already listed. Other items that were already in permitted_enctypes are left alone and not removed.

- true — replace the targeted krb5.conf parameters so they match exactly what is specified in centrifydc.conf.

  Items from centrifydc.conf are added, if they were not already listed. Other items that were already in permitted_enctypes, and not in centrifydc.conf, are removed.

To apply changes to this parameter, either restart adclient or ensure the group policy is set as follows: **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings > Control if strictly enforce the permitted_encTypes**.

adclient.krb5.principal

This configuration parameter specifies whether SAM account names or user principal names (UPNs) are used as the principal in Kerberos tickets. Supported values are sam and upn.

For example:

adclient.krb5.principal: sam

The default value is sam.

If you set this parameter to upn and no UPN is available, the sAMAccountName attribute with the format SAMAccountName@DomainName is used.

In MIT Kerberos environments, however, the UPN is used even if this parameter is set to sam.

adclient.krb5.send.netbios.name

This configuration parameter specifies whether the Delinea Agent sends the host computer's NetBIOS name (the computer's pre-Windows 2000 name) together with the host computer's IP address (or addresses) when the agent makes a Kerberos authentication request that includes IP addresses. The NetBIOS name appears in the domain controller log on the host Windows server and helps identify the computer making the request.

If this parameter is set to true, the agent sends the NetBIOS name. If set to false, the agent does not send the NetBIOS name.

This parameter has no effect unless adclient.krb5.use.addresses is set to true.

If this parameter is not defined in the configuration file, its default value is true.

adclient.krb5.service.principals

This configuration parameter specifies additional service principals for entries in the Kerberos key table. The key table is populated by default with the service principals host and ftp cifs.

This parameter's value must be one or more principal service names, separated by a space or by a comma. For example:

adclient.krb5.service.principals: ldap nfs

If this parameter is not defined in the configuration file, no additional principal names are added to the Kerberos key table.

adclient.krb5.tkt.encryption.types

This configuration parameter specifies the types of encryption that can be presented to the server in the TGT when the computer is requesting service tickets.

The parameter value must be one or more encryption types, separated by a space. For example:

adclient.krb5.tkt.encryption.types: arcfour-hmac-md5 des-cbc-md5

If this parameter is not defined in the configuration file, the default encryption types permitted are:

- Windows 2000 server and Windows Server 2003: arcfour-hmac-md5, des-cbc-md5, and des-cbc-crc.
- Windows Server 2008 domain functional level supports these additional types:

  aes128-cts and aes256-cts.
  Although you can specify these types in an environment other than 2008 domain functional level, they are not useful and may cause extra network round trips during the authentication process.

adclient.krb5.tkt.encryption.type.strict

The adclient.krb5.tkt.encryption.type.strict parameter controls whether to replace the encryption types set in default_tgs_enctypes and default_tkt_enctypes in krb5.conf with the encryption types specified in adclient.krb5.tkt.encryption.types in centrifydc.conf.

- When adclient.krb5.tkt.encryption.type.strict is false (default), then:

  The encryption types listed adclient.krb5.tkt.encryption.types in centrifydc.conf are added to the list of encryption types in default_tgs_enctypes and default_tkt_enctypes in krb5.conf.

  This only ensures that what is specified in centrifydc.conf is present in krb5.conf. It does not remove unknown items.

- When adclient.krb5.tkt.encryption.type.strict is set to true, then:

  The encryption types listed in adclient.krb5.tkt.encryption.types in centrifydc.conf replace the encryption types specified in the settings, default_tgs_enctypes and default_tkt_enctypes, in krb5.conf.

  The permitted encryption types in krb5.conf exactly match the permitted encryption types in centrifydc.conf. Extra or unknown encryption types are removed.

Example:

adclient.krb5.tkt.encryption.type.strict: false

- false — Default is false. No change in behavior. default_tgs_enctypes and default_tkt_enctypes are updated from the centrifydc.conf file.

  Items from centrifydc.conf are added, if they were not already listed. Other items that were already in default_tgs_enctypes and default_tkt_enctypes are left alone and not removed.

- true — Replace the targeted krb5.conf parameters so they match **exactly** what is specified in centrifydc.conf.

  Items from centrifydc.conf are added, if they were not already listed. Other items that were already in default_tgs_enctypes and default_tkt_enctypes, and not in centrifydc.conf, are removed.

To apply changes to this parameter, either restart adclient or ensure the group policy is set as follows: **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings > Control if strictly enforce the encTypes**.

adclient.krb5.use.addresses

This configuration parameter controls whether the Delinea Agent should send the host computer's local IP address (or addresses) to the Windows domain controller as part of a Kerberos authentication request. When set to true, the agent sends the IP addresses; when set to false, the agent does not send the IP addresses.

When the agent sends the host computer's IP address with a Kerberos request, the IP address appears in the Windows event logs associated with the request.

This configuration parameter works with the parameters adclient.krb5.extra_addresses and adclient.krb5.send.netbios.name. Use the first of these two parameters to add additional IP addresses to the host computer's IP address (useful in networks using NAT). Use the second to add the host computer's NetBIOS name to the IP address (or addresses) (useful for identifying the requesting computer in event logs).

If adlcient.krb5.use.addresses is set to false, neither of these two parameters has any effect because the agent does not send addresses with an authentication request.

> **Note**: This configuration parameter sets the Kerberos configuration parameter noaddresses in krb5.conf. Setting adclient.krb5.use.addresses to true sets noaddresses to false; setting adclient.krb5.use.addresses to false sets noaddresses to true.

If adclient.krb5.use.addresses is not defined in the configuration file, its default value is false.

fips.mode.enable

This configuration parameter specifies whether Kerberos uses the algorithms in the FIPS 140-2 compliant library to sign and seal messages. See fips.mode.enable for the description.

krb5.cache.clean

This configuration parameter specifies whether Kerberos credentials in the cache should be deleted when a user logs out. By default, credentials stored in the Kerberos cache that belong to users who are not logged in are periodically deleted.

To keep the credentials available in the cache use this parameter to turn off the cache clean process entirely. Alternatively, use the krb5.cache.clean.exclusion to turn off cache cleaning for specific users.

This configuration parameter allows you to control this operation specifically for zone users or for all users.

The parameter value must be one of the following valid settings:

- off to turn off the deletion of the credentials cache for all users.
- cdc to remove all of the /tmp/krb5cc* files created by the agent (adclient) that belong to any user not found in the utmp database (that is, the user has logged out).
- all to remove all of the /tmp/krb5cc* files that belong to any user not found in the utmp database. This setting removes files created by the agent (adclient), telnet, and openssh.

For example, to remove the credentials cache for all users when they log out:

krb5.cache.clean: all

The default value for this parameter is cdc.

krb5.cache.clean.exclusion

This configuration parameter specifies a list of users whose credentials in the Kerberos cache will *not* be deleted during a periodic Kerberos cache clean-out of unlogged-in users.

Each user is specified by the user's UNIX name. Separate the names in the list using a comma.

For example, to specify that three users be excluded from periodic credential clean-up:

krb5.cache.clean.exclusion: admin,paula,jeffrey

This parameter is useful in a batch processing environment where a logged-out user may leave behind running processes that require Kerberos credentials. It allows some users' credentials to remain for processes while cleaning out all other users' credentials.

The default value for this parameter is empty.

krb5.cache.clean.force.max

This configuration parameter controls whether adclient deletes credentials from the Kerberos cache if they are the specified number of days old.

If you activate this parameter, the credentials will be cleared for all users whether or not they are logged on, have active processes running, or are specified in the following lists:

krb5.cache.clean.exclusion
krb5.cache.infinite.renewal.batch.users
krb5.cache.infinite.renewal.batch.groups

For example, to force adclient to clear the cache of credentials that were authenticated 6 days previously:

krb5.cache.clean.force.max: 6

The default value for this parameter is 0, which means that this configuration parameter will not clear the credential cache for any users.

krb5.cache.clean.interval

This configuration parameter specifies how frequently in minutes to check the Kerberos cache for credentials that belong to users who are not logged on. If the user is not logged on, the credentials are deleted.

The parameter value should be a positive integer. Setting this parameter to zero disables periodic clean-up of the cache.

For example, to set the clean-up interval to 5 minutes:

krb5.cache.clean.interval: 5

The default value for this parameter deletes the credential cache for users who have logged off every one minute.

krb5.cache.infinite.renewal

This configuration parameter specifies whether you want user credentials to be automatically reissued when they expire. The parameter value can be set to true or false. If you set this parameter to true, the agent keeps a hash of the user's password in memory indefinitely. If you set this parameter to false, a user's credentials periodically expire and the user must be re-authenticated by re-entering a valid password.

If you set this parameter to true, user credentials are automatically reissued, as needed, as long as the adclient process continues to run even if the computer is disconnected from Active Directory. If you stop or restart adclient, however, the user's password hash is removed from memory. After stopping or restarting adclient, users must be re-authenticated by logging on with a valid user name and password.

The default parameter value is false. For example:

krb5.cache.infinite.renewal: false

krb5.cache.infinite.renewal.batch.groups

This configuration parameter specifies a list of Active Directory groups whose members' Kerberos credentials require infinite renewal even after the users have logged out.

Requirements to use this parameter:

- Specified groups must be Active Directory groups.
- Groups do not need to be zone enabled.
- To have their credentials automatically renewed, users in the group must:
  - Be zone enabled (that is, mapped users are not supported).
  - Log into the desired system once using the Account Password.

You must use the following format to specify group names:

SamAccountName@domain

For example:

krb5.cache.infinite.renewal.batch.groups: test_group_sam@example.com

By default, this parameter does not list any groups.

You can also use group policy to set this parameter.

krb5.cache.infinite.renewal.batch.users

This configuration parameter specifies a list of users whose Kerberos credentials require infinite renewal even after the users have logged out.

Requirements to use this parameter:

- The users must be zone enabled (that is, mapped users are not supported).
- The users must log into the desired system once using the Account Password.

You can use any of the following formats to specify user names:

unixName
userPrincipleName
SamAccountName
SamAccountName@domain

For example:

krb5.cache.infinite.renewal.batch.users: test_user, test_user@example.com, test_user_sam, test_user_sam@example.com

By default, this parameter does not list any users.

You can also use group policy to set this parameter.

krb5.cache.renew.exclusion

This configuration parameter specifies a list of UNIX users for whom you don't want adclient to automatically renew their Kerberos credential caches. This parameter is useful in situations where you need to directly manage certain users' Kerberos caches.

Specify each user by the user's UNIX name. Separate the names in the list using a comma.

For example, to specify that adclient doesn't renew these three users' Kerberos credential caches:

krb5.cache.renew.exclusion: admin,paula,jeffrey

Alternatively, you can using the file: keyword to specify a separate file that contains UNIX user names.

For example:

krb5.cache.renew.exclusion: file:/etc/centrifydc/renew.exclude

You can put a UNIX user name in each single line, and be sure to run the adreload command after modifying the file to have the changes take effect.

The default value for this parameter is empty.

krb5.cache.renew.interval

This configuration parameter specifies, in hours, how often to renew the Kerberos credentials stored in the cache for users who have logged on successfully. Because Kerberos tickets expire after a set period of time, you can use this configuration parameter to periodically renew the existing Kerberos ticket to keep existing credentials valid.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. A value of zero disables renewal. For example, to set the renewal interval to 8 hours:

krb5.cache.renew.interval = 8

If this parameter is not defined in the configuration file, its default value is 4 hours. The default value of 4 hours allows two attempts at renewal over a typical Kerberos ticket lifespan of 10 hours. If possible, you should allow enough time for at least two renewal attempts if you reset the value to something other than the 4-hour default.

krb5.conf.plugins.ccselect.disable

This configuration parameter controls whether adclient disables the Kerberos built-in ccselect plugins.

If you set this parameter to false, the plugins will not be disabled.

For example,

krb5.conf.plugins.ccselect.disable: false

By default, this parameter is set to true, and the built-in ccselect plugins are disabled.

You can also set this parameter using group policy.

krb5.cache.type

This configuration parameter specifies the type of Kerberos credential cache that the agent (adclient) creates when an Active Directory user logs in. The parameter value can be set to FILE or KCM.

> **Note**: The use of in-memory credential caches such as KCM is not supported on Mac OS X computers. In Mac OS X environments, credential caches are file-based, and setting this parameter has no effect.

If you set this parameter to FILE, the agent creates a file-based credential cache for each Active Directory user in /tmp when the user logs in. A file-based credential cache persists until the file is deleted.

If you set this parameter to KCM, the agent creates an in-memory credential cache for each Active Directory user when the user logs in. The Centrify-KCM service, run as root, manages in-memory credential caches. When the agent, adclient, starts up, if the parameter is set to KCM, adclient starts the KCM service. If you change the parameter from FILE to KCM while adclient is running, adclient starts the KCM service the next time it is forced to reload configuration parameters, for example, if you run the adreload command or if a user opens a new session.

Setting this parameter affects new users only — not users who have already logged in. For example, if you change from a file-based, to an in-memory credential cache, the agent will continue to use the file-based credential cache for any user who was logged in at the time of the change. If a logged in user opens a new session, or a new user logs in, the agent will use an in-memory cache for them.

An in-memory credential cache ends as soon as the Centrify-KCM service is stopped.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The default parameter value is FILE, which specifies a file-based credential cache. To specify an in-memory credential cache, set the value to KCM. For example:

krb5.cache.type: KCM

krb5.conf.k5login.directory

Use this policy to specify an alternative location for user .k5login files.

If specified, this string value will be used for the k5login_directory in the [libdefaults] stanza in krb5.conf and the user's .k5login file will be named as <k5login_directory>/<unix_name>.

For security reasons the specified directory should be owned by root and writeable by root only. If the directory does not exist, adclient will create it.

krb5.conf.kcm.socket.path

The krb5.conf.kcm.socket.path parameter specifies an alternate socket path for the KCM server. It applies when krb5.cache.type is KCM.

This is useful, as it allows you to configure an alternative kcm socket path, for example, /var/centrifydc. Using an alternative socket path then allows the socket to be shared between docker hosts and docker containers. It requires adreload after a change in value.

- When the parameter is an empty string (default), the default path /var/run/.centrify-kcm-socket is used.
- When the parameter is set to an non-empty string AND krb5.conf.kcm.socket.path.secure.usable.check is false, then this socket path is used without secure and usable check.
- When the parameter is set to an non-empty string AND krb5.conf.kcm.socket.path.secure.usable.check is true, then the configured socket path is checked to see if it is valid:
    - If the socket path is valid, this configured socket path is used.
    - If the socket path is not valid, the default socket path, /var/run/.centrify-kcm-socket, is used.

To change the socket path:

1. In centrifydc.conf, set krb5.conf.kcm.socket.path to a valid path.
2. If the configured kcm socket path is not secure, but you still want to use it, ensure the parameter, krb5.conf.kcm.socket.path.secure.usable.check, is false.
3. Run adreload.

krb5.conf.kcm.socket.path.secure.usable.check

The krb5.conf.kcm.socket.path.secure.usable.check parameter specifies whether to perform a secure and usable check on the alternate socket path for the KCM server. This parameter works in conjunction with krb5_conf_kcm_socket_path. Options are:

- false — Default. No action taken.
- true — If krb5.conf.kcm.socket.path is configured, then krb5.conf.kcm.socket.path.secure.usable.check checks the specified directory.

A socket path is valid when it meets the following criteria:

- the parent directory exists
- the parent directory is not a symlink
- the parent directory is writable by root only
- the socket path does not exist, or it exists but it is not directory

krb5.config.update

This configuration parameter specifies, in hours, how frequently the agent updates the Kerberos configuration file.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If adclient.krb5.autoedit is set to false, this parameter has no effect. If adclient.krb5.autoedit is set to true, this parameter value must be a positive integer. For example, to set the update interval to 8 hours:

krb5.config.update: 8

If this parameter is not defined in the configuration file, its default value is 8 hours.

krb5.forcetcp

This configuration parameter specifies whether to allow Kerberos requests to use UDP or to force all Kerberos requests to use TCP.

If krb5.forcetcp is set to false, Kerberos requests may use UDP. If krb5.forcetcp is set to true, all Kerberos requests use TCP only.

In most cases, you set this configuration parameter using group policy.

You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If this parameter is not defined in the configuration file, its default value is true.

krb5.forwardable.user.tickets

This configuration parameter specifies whether you want the agent to create forwardable Kerberos user tickets. Creating a forwardable ticket allows a user's logon ticket to be sent to another computer and used to access to additional systems and resources. For example, if a user logs on and is authenticated on one computer, then uses a Kerberized telnet session to connect to a second computer, a forwarded ticket allows the user to access to additional Kerberized resources from that second computer without separate authentication.

In most environments, forwarding user tickets is a safe practice. However, if you do not want tickets to be forwarded, you can use this parameter to prevent the agent from creating forwardable tickets.

The parameter value should be 1 is you want to allow ticket forwarding or 0 is you want to prevent ticket forwarding. For example, if you want the agent to create forwardable user tickets:

krb5.forwardable.user.tickets: 1

If this parameter is not defined in the configuration file, its default value is 1 (yes).

krb5.pac.validation

This configuration parameter specifies whether or not to verify that the user's PAC (Privilege Authorization Certificate) information is from a trusted KDC (Key Distribution Center) so as to prevent what's referred to as a "silver ticket" attack.

When performing credential verification, a service ticket is fetched for the local system. After the credential is verified, the local system uses the PAC information in the service ticket.

This setting take effect when krb5.verify.credentials is enabled or when DirectControl is using the user's PAC from a service ticket. This setting does not apply to retrieving the PAC by way of the S4U2Self protocol.

There are 3 possible values for krb.pac.validation:

- disabled (default): NO PAC validation will be done at all.
- enabled: If PAC Validation fails, the PAC information is used and the user login is allowed.
- enforced: If PAC Validation fails, the PAC information is discarded and the user login is denied.

Setting this parameter to enabled or enforced will have significant impact on the user login and user's group fetch performance.

For example:

krb5.pac.validation: disabled

If this parameter is not defined in the configuration file, its default value is disabled.

krb5.permit.dns.spn.lookups

This configuration parameter specifies whether you want to permit the agent to look up service principal names (SPN) using DNS. In most cases, you should set this parameter to false to ensure the security of the system. You should only set this configuration parameter to true if you can safely rely on DNS for security and want to use programs that use the Delinea Kerberos libraries to access a computer using an IP address or localhost.

For example:

krb5.permit.dns.spn.lookups: false

If this parameter is not defined in the configuration file, its default value is false.

krb5.sso.block.local_user

This configuration parameter specifies whether single sign-on (SSO) is permitted for local users, or if only zone-enabled Active Directory users are allowed to log in through SSO.

By default, this parameter is set to true, and the user UNIX name is checked against the nss.ignore.user list. If the UNIX name is in the list, the user is considered a local user, and SSO is not allowed. In this situation, the user must enter the local user password to log in.

If this parameter is set to false, local users are allowed to log in through SSO.

For example:

krb5.sso.block.local_user: true

krb5.sso.ignore.k5login

This configuration parameter specifies whether the k5login module should ignore .k5login for SSO.

The default value is false.

krb5.support.alt.identities

This configuration parameter specifies whether the agent uses the Kerberos altSecurityIdentities name for user authentication (true) or not (false) instead of the Windows user name, regardless of which names are supplied.

Using altSecurityIdentities for authentication works as long as the alternate name is always used or the passwords are synchronized, and if the third-party key distribution center (KDC) is reachable. If these two conditions aren't met, you can disable the feature by setting this parameter to false. In that case, the agent uses only Windows to authenticate the user and ignores any Kerberos altSecurityIdentities.

For example:

krb5.support.alt.identities: false

If this parameter is not defined in the configuration file, its default value is true.

krb5.unique.cache.files

This configuration parameter specifies whether to generate a unique ticket cache file name for each Kerberos authentication for a given user (except the first). The unique ticket cache file name takes the following form:

krb5cc_cdc<uid>_XXXXXX

The <uid> is the users Unix ID, and the XXXXXX is a unique set of characters (i.e. krb5cc_cdc512_u0PSdt). This allows a given user to log on more than once, without subsequent logoffs interfering with other logon instances.

If this parameter is set to false, the ticket cache filename takes the following form:

krb5cc_<UID>

With this parameter set to false, old versions of the ticket cache file are overwritten. If a user logs in twice, the first logout causes the file to be deleted, leaving the other logon instance without a credential cache.

The environment variable KRB5CCNAME is populated with the generated name.

The default value is true, except on macOS where it is false.

krb5.use.kdc.timesync

This configuration parameter enables Kerberos to automatically correct for a time difference between the system clock and the clock used by the KDC. You only need to set this parameter if your system clock is drifting and the system is not using NTP and adclient SNTP settings.

In most cases, you set this configuration parameter using group policy.

You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

For example:

krb5.use.kdc.timesync: true

If this parameter is not defined in the configuration file, its default value is false.

krb5.verify.credentials

This configuration parameter specifies whether to perform a spoofing check to verify a TGT for the local system.

By default, the agent verifies a user's TGT by retrieving and verifying a service ticket for the local system. This check is done to prevent a well-known attack (the Zanarotti or screen-saver attack) whereby a rogue KDC could respond to the agent's request for the user's TGT.

However, the spoofing check can be time consuming, so you can set this parameter to false to disable the spoofing check and significantly improve authentication performance.

For example, to disable the check:

krb5.verify.credentials: false

If this parameter is not defined in the configuration file, the default value is true.

krb5.udp.preference.limit

This configuration parameter sets the maximum size packet that the Kerberos libraries will attempt to send over a UDP connection before retrying with TCP. If the packet size is larger than this value, only TCP will be tried. If the value is set to 1, TCP will always be used. The hard UDP limit is 32700. Values larger than this are ignored and the UDP hard limit is enforced.

This key only takes effect if krb5.forcetcp is set to false.

If krb5.forcetcp is true, and the agent is managing the krb5.conf file, it will set udp_preference_limit = 1, so that the Kerberos libraries will always use TCP.

In most cases, you set this configuration parameter using group policy to set a specific value.

You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If this parameter is not defined in the configuration file, the default value is 1465; for example:

krb5.udp.preference.limit:1465

Customizing PAM-Related Configuration Parameters

This section describes the configuration parameters that affect the operation of PAM related activity on the local host computer.

Configuring PAM-related parameters on IBM AIX computers

On IBM AIX computers, the PAM configuration parameters described in this chapter apply to the AIX Loadable Authentication Module (LAM) or to the PAM interface. If you have configured the AIX computer to use the PAM interface, the configuration parameters apply to the PAM settings. If the AIX computer is configured to use the LAM interface, the parameters configure LAM settings, as applicable. For more information about AIX-specific configuration parameters, see the Customizing AIX configuration parameters.

Controlling access to AIX computers

On most computers, the predefined login-all PAM access right is required to allow users who are assigned the UNIX Login role to log on and use PAM-enabled applications in the zones they have permission to access. However, if you have AIX computers that are configured to use the Loadable Authentication Module (LAM) instead of PAM in a zone, users will be able to log on even if they have not been assigned the UNIX Login role. In addition, if you define your own custom PAM access rights, those rights will not be applicable on AIX computers that use the LAM interface.

To prevent users from logging on to or using unauthorized applications on AIX computers is a zone, you can explicitly allow or deny access to specific users and groups through configuration parameters or group policies or change the configuration of your AIX computers to use the more commonly supported Pluggable Authentication Module (PAM) interface. For more information about controlling access, see Enforcing access rights on AIX computers.

Explicitly allowing and denying access

If you have AIX computers that use the Loadable Authentication Module (LAM) interface, you cannot use the predefined login-all PAM access right or custom PAM access rights to authorize who can log on and who can use specific applications. Therefore, the default UNIX Login role does not apply on AIX computers that use the LAM interface. If you are primarily concerned with who can log on to those computers, you can use the pam.allow.groups, pam.allow.users, or both parameters to explicitly specify the groups and users who can log on to AIX computers that use the LAM interface. All other groups and users—including those assigned the UNIX Login role—will be denied access. Alternatively, you can use the pam.deny.users, pam.deny.groups, or both parameters to explicitly specify the users and groups who are not allowed to log on.

Changing the configuration of AIX computers

By default, AIX computers are configured to use the Loadable Authentication Module (LAM) instead of the Pluggable Authentication Module (PAM) subsystem. If you want to be able to use the default or custom PAM access rights to authorize access to specific applications, you might want to reconfigure your AIX computers to use the PAM interface instead of the LAM interface. If you choose to reconfigure AIX computers, you should also be sure to replace the OpenSSH package for LAM with the OpenSSH for PAM and thoroughly test your applications.

pam.account.conflict.both.mesg

This configuration parameter specifies the message displayed if both user name and user ID conflicts are detected during login; that is, there are two local account conflicts. For example, a local user (user2) and the Active Directory user (user1) have the same UID (10001) but different user names, and another local account has the same user name (user1) as the Active Directory user but has a different UID value (10002):

```
user1 10001 #AD User
user1 10002 #local user
user2 10001 #local user
```

When the message is displayed, the %s token in the message string is replaced with the name of the first conflicting local account, and the %d token is replaced with the UID of the second conflicting local account. The message string you define must contain exactly one %s token and exactly one %d token, in that order, and no other string replacement (%) characters.

For example:

```
pam.account.conflict.both.mesg: \
Accounts with conflicting name (%s) and UID (%d) exist locally
```

For more information about displaying a warning when local conflicts are detected, see pam.uid.conflict.

pam.account.conflict.name.mesg

This configuration parameter specifies the message displayed if a user name conflict is detected during login; that is, if there is a local user with the same name but a different UID than the Active Directory user logging on; for example,

user1 10001 #local user
user1 10002 #AD user

When the message is displayed, the %s token in the message string is replaced with the name of the conflicting local account. The message string you define must contain exactly one %s token, and no other string replacement (%) characters.

For example:

pam.account.conflict.name.mesg: \
Accounts with conflicting name (%s) exist locally

For more information about displaying a warning when local conflicts are detected, see pam.uid.conflict.

pam.account.conflict.uid.mesg

This configuration parameter specifies the message displayed if a user identifier (UID) conflict is detected during login; that is, if there is a local user with a different user name but the same UID as the Active Directory user logging on. For example:

user1 10001 #local user
user2 10001 #AD user

When the message is displayed, the %d token is replaced with the UID of the conflicting local account. The message string you define must contain exactly one %d token, and no other string replacement (%) characters.

For example:

pam.account.conflict.uid.mesg: \
Account with conflicting UID (%d) exists locally

For more information about displaying a warning when local conflicts are detected, see pam.uid.conflict.

pam.account.disabled.mesg

This configuration parameter specifies the message displayed if a user attempting to log on is denied access because the user's account has been disabled in Access Manager or Active Directory Users and Computers.

For example:

pam.account.disabled.mesg: Account cannot be accessed at this time. Please contact your system administrator.

pam.account.expired.mesg

This configuration parameter specifies the message displayed if a user attempting to log on is denied access because the user's account has expired.

For example:

pam.account.expired.mesg:
Account cannot be accessed at this time. Please contact your system administrator.

pam.account.locked.mesg

This configuration parameter specifies the message displayed if a user account is locked because of too many failed login attempts.

For example:

pam.account.locked.mesg: Account locked

> **Note**: These messages may not be displayed depending on the login method, the daemon version, or the version of the operating system. (Ref: CS-16710c)

pam.adclient.down.mesg

This configuration parameter specifies the message displayed during password change if user is a local UNIX user that's mapped to an Active Directory account, and the Delinea Agent (adclient) is not accessible.

For example:

pam.adclient.down.mesg: (Unable to reach Active Directory - using local account)

In most cases, you set this configuration parameter by selecting **Enabled** and specifying the message to be displayed.

pam.allow.groups

This configuration parameter specifies the groups allowed to access PAM-enabled applications. When this parameter is defined, only the listed groups are allowed access. All other groups are denied access.

> **Note**: This parameter does not support cross-forest groups. (Ref: CS-18659a)

If you want to use this parameter to control which users can log in based on group membership, the groups you specify should be valid Active Directory groups, but the groups you specify do not have to be enabled for UNIX. Local group membership and invalid Active Directory group names are ignored.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you use this parameter to control access by group name, the agent checks the Active Directory group membership for every user who attempts to use PAM-enabled applications on the host computer.

When a user attempts to log on or access a PAM-enabled service, the pam_centrifydc module checks with Active Directory to see what groups the user belongs to. If the user is a member of any Active Directory group specified by this parameter, the user is accepted and authentication proceeds. If the user is not a member of any group specified by this parameter, authentication fails and the user is rejected.

The parameter's value can be one or more group names, separated by commas, or the file: keyword and a file location. For example, to allow only members of the administrators, sales, and engineering groups in Active Directory to log in:

pam.allow.groups: administrators,sales,engineering

You can use the short format of the group name or the full canonical name of the group.

To enter group names with spaces, enclose them in double quotes; for example:

pam.allow.groups: "domain admins",sales,"domain users"

To specify a file that contains a list of the groups allowed access, type the path to the file:

pam.allow.groups: file:/etc/centrifydc/groups.allow

If no group names are specified, no group filtering is performed.

If you make changes to this parameter, you should run adflush to clear the cache to ensure your changes take effect.

Specifying group names for computers joined to Auto Zone

If a computer is configured to use the Auto Zone instead of a specific zone, you should specify group names using the format defined by the auto.schema.name.format parameter. For example the auto.schema.name.format parameter can be set to the following:

- SAM (default) uses the samAccountName attribute for the group—web_qa
- SAM@domainName uses the samAccountName@domain_name format—web_qa@acme.com
- NTLM uses the NTLM format and separator defined for adclient.ntlm.separators—acme.com+web_qa

You can look in the centrifydc.conf configuration file for the value of auto.schema.name.format, or run adedit or adquery commands to see the UNIX name for any group. For example, to see the UNIX name for the Web_qa Active Directory group when the auto.schema.name.format parameter is set to SAM, you can execute a command similar to this to return the UNIX group profile name:

```
adquery group -n web_qa
webqa.us
```

pam.allow.override

This configuration parameter is used to override authentication through Active Directory to ensure the root user or another local account has permission to log on when authentication through Active Directory is not possible, when there are problems running the adclient process, or when there are network communication issues.

When you specify a user account for this parameter, authentication is passed on to a legacy authentication mechanism, such as /etc/passwd. You can use this parameter to specify an account that you want to ensure always has access, even if communication with Active Directory or the adclient process fails. For example, to ensure the local root user always has access to a system even in an environment where you have enabled root mapping, you can specify:

pam.allow.override: root

To log in locally with the override account, you must specify the local user name and password. However, because the account is mapped to an Active Directory account, you must append @localhost to the user name. For example, if you have specified root as the override account and are using root mapping, you would type root@localhost when prompted for the user name. You can then type the local password for the root account and log in without being authenticated through Active Directory.

> **Note**: If you are mapping the root user to an Active Directory account and password, you should set this parameter to root or to a local user account with root-level permissions (UID 0), so that you always have at least one local account with permission to access system files and perform privileged tasks on the computer even if there are problems with the network connection, Active Directory, or the adclient process.

>**Note**: If you are using a Solaris machine with the Name Switch Cache Daemon (NSCD) running, you will not be able to log in as an override user using \<*username*\>@localhost. (Ref: CS-29816c)

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

>**Note**: The pam.allow.override configuration parameter is not supported on AIX computers. This is because using the user name with the suffix @localhost is not supported on AIX. The LAMGetEntry call that is used to get user information and extended attribute information does not support login name changes. So, the login fails as there is no way to find the user or authenticate the user. There is no equivalent setting for AIX computers. (Ref: CS-33506a)

pam.allow.password.change

This configuration parameter specifies whether users who log in with an expired password should be allowed to change their password. You can set this parameter to true or false and use it in conjunction with the pam.allow.password.expired.access parameter to control access for users who attempt to log on with an expired password.

If both this parameter and pam.allow.password.expired.access are set to true, users logging on with an expired password are allowed to log on and are prompted to change their password.

If the pam.allow.password.expired.access parameter is set to true, but this parameter is set to false, users logging on with an expired password are allowed to log on but are not prompted to change their password and the message defined for the pam.allow.password.change.mesg parameter is displayed.

If both this parameter and pam.allow.password.expired.access are set to false, users who attempt to log on with an expired password are not allowed to log on or change their password and the message defined for the pam.allow.password.change.mesg parameter is displayed.

For example, to allow users with expired passwords to change their password:

pam.allow.password.change: true

pam.allow.password.change.mesg

This configuration parameter specifies the message displayed when users are not permitted to change their expired password because the pam.allow.password.change parameter is set to false.

For example:

pam.allow.password.change.mesg: Password change not permitted

pam.allow.password.expired.access

This configuration parameter specifies whether users who log in with an expired password should be allowed access. You can set this parameter to true or false and use it in conjunction with the pam.allow.password.change parameter to control access for users who attempt to log on with an expired password.

If this parameter is set to true, users logging on with an expired password are allowed to log on, and either prompted to change their password if the pam.allow.password.change parameter is set to true, or notified that they are not allowed to change their expired password if the pam.allow.password.change parameter is set to false.

If this parameter is set to false, users logging on with an expired password are not allowed to log on and the message defined for the pam.allow.password.expired.access.mesg parameter is displayed.

For example, to allow users with expired passwords to log on:

pam.allow.password.expired.access: true

pam.allow.password.expired.access.mesg

This configuration parameter specifies the message displayed when users are not permitted to log on with an expired password because the pam.allow.password.expired.access parameter is set to false.

For example:

pam.allow.password.expired.access.mesg: Password expired - access denied

pam.allow.users

This configuration parameter specifies the users who are allowed to access PAM-enabled applications. When this parameter is defined, only the listed users are allowed access. All other users are denied access.

If you want to use this parameter to control which users can log in, the users you specify should be valid Active Directory users that have a valid UNIX profile for the local computer's zone. If you specify local user accounts or invalid Active Directory user names, these entries are ignored.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you specify one or more users with this parameter, user filtering is performed for all PAM-enabled applications on the host computer.

When a user attempts to log on or access a PAM-enabled service, the pam_centrifydc module checks the users specified by this parameter to see if the user is listed there. If the user is included in the list, the user is accepted and authentication proceeds. If the user is not listed, the user is rejected.

The parameter value can be one or more user names, separated by commas, or the file: keyword and a file location. For example:

pam.allow.users: root,joan7,bbenton
pam.allow.groups: administrators,sales,engineering

You can use the short format of the user name or the full canonical name of the user.

To enter user names with spaces, enclose them in double quotes; for example:

pam.allow.users: "sp1 user@acme.com",joan@acme.com,"sp2 user@acme.com"

To specify a file that contains a list of the users allowed access, type the path to the file:

pam.allow.users: file:/etc/centrifydc/users.allow

If no user names are specified, then no user filtering is performed.

If you make changes to this parameter, you should run adflush to clear the cache to ensure

Specifying user names for computers joined to Auto Zone

If a computer is configured to use the Auto Zone instead of a specific zone, you should specify user names using the format defined by the auto.schema.name.format parameter. For example the auto.schema.name.format parameter can be set to the following:

- SAM (default) uses the samAccountName attribute for the user—jcool
- SAM@domainName uses the samAccountName@domain_name format—jcool@acme.com
- NTLM uses the NTLM format and separator defined for adclient.ntlm.separators

You can look in the centrifydc.conf configuration file for the value of auto.schema.name.format parameter or run adedit or adquery commands to see the UNIX name for any user. For example, to see the UNIX name for the jcool Active Directory user when the auto.schema.name.format parameter is set to SAM, you can execute a command similar to this to return the UNIX user profile name:

adquery user -n jcool

pam.auth.create.krb5.cache

This configuration parameter specifies whether PAM creates the Kerberos user credential cache. A value of true specifies that the Kerberos user credential cache is created. A value of false specifies that the Kerberos user credential cache is not created. The default value is true. For example:

pam.auth.create.krb5.cache: true

- When this parameter is set to false, the Kerberos user credential cache is not created, and any attempt to perform an SSO operation will fail.
- The Kerberos user credential cache can be file-based or it can be a KCM in-memory cache, depending on the krb5.cache.type setting (see krb5.cache.type.
- This parameter is also controlled by group policy.

pam.auth.failure.mesg

This configuration parameter specifies the message displayed during a password change if the user enters an incorrect old password.

For example:

pam.auth.failure.mesg: Password authentication failed

pam.config.program.check

This configuration parameter specifies a list of extra PAM configuration files that are in the pam.d directory and that the authentication service updates (in addition to the standard PAM configuration files, such as /pam.d/system-auth, /pam.d/common-auth and so forth).

The default list is as follows:

pam.config.program.check: ftp,pure-ftpd,vsftpd,wu-ftpd,dzdo,sasauth

pam.create.k5login

This configuration parameter specifies whether the .k5login file should be created automatically in the user's home directory. This file is used to enable Kerberos authentication and single sign-on in PAM-aware applications.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value can be true or false. If set to true, the agent will create the .k5login file in the user's home directory.

For example:

pam.create.k5login: true

pam.deny.change.shell

This configuration parameter specifies whether a user who is denied access, for example, because they are listed as a user in the pam.deny.user or are not listed in the pam.allow.user parameter, should have their shell set to the shell defined by the nss.shell.nologin parameter. The parameter value can be set to true or false.

If set to true, this parameter adds an extra level of security by ensuring that the zone user who is denied access cannot obtain any shell access, even if authenticated through Kerberos, SSH, or some other non-PAM related method. If this parameter is set to false, the denied user's shell is not changed and so may be able to access the system.

Because of the potential security issue, the default value for this parameter is true. However, since group lookups can be time-consuming for simple NSS queries, you can set this parameter to false to prevent the agent from changing the user's shell when denied access.

For example, to leave the user's shell unchanged when denied access, set this parameter to false.

pam.deny.change.shell: false

pam.deny.groups

This configuration parameter specifies the groups that should be denied access to PAM-enabled applications. When this parameter is defined, only the listed groups are denied access. All other groups are allowed access.

> **Note**: This parameter does not support cross-forest groups. (Ref: CS-18659a)

If you want to use this parameter to control which users can log in based on group membership, the groups you specify should be valid Active Directory groups, but the groups you specify do not need to be enabled for UNIX. Local group membership and invalid Active Directory group names are ignored.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

When a user attempts to log on or access a PAM-enabled service, the pam_centrifydc module checks with Active Directory to see which groups the user belongs to. If the user is a member of any Active Directory group specified by this parameter, the user is denied access and authentication fails. If the user is not a member of any group specified by this parameter, authentication succeeds and the user is logged on.

The parameter's value can be one or more group names, separated by commas or spaces, or the file: keyword and a file location. For example, to prevent all members of the vendors and azul groups in Active Directory from logging on:

pam.deny.groups: vendors,azul

You can use the short format of the group name or the full canonical name of the group.

To enter group names with spaces, enclose them in double quotes; for example:

pam.deny.groups: "domain admins",sales,"domain users"

To specify a file that contains a list of the groups that should be denied access:

pam.deny.groups: file:/etc/centrifydc/groups.deny

> **Note**: If a computer is configured to use Auto Zone without a zone, enter group names in the format specified by the auto.schema.name.format parameter:

- SAM (samAccountName — this is the default); for example: finance_admins
- samAccountName@domain_name; for example: finance_admins@acme.com
- NTLM; for example: acme.com+finance_admins

> **Note**: You can look in the centrifydc.conf configuration file for the value of auto.schema.name.format, or run adquery group -n to see the UNIX name for any group. For example, to see the UNIX name for the Finance_Admins group (and SAM, the default, is set for auto.schema.name.format), execute the following command, which returns the UNIX name as shown:

```
[root]#adquery group -n Finance_Admins
finance_admins
```

If this parameter is not defined in the configuration file, no group filtering is performed.

> **Note**: If you make changes to this parameter, you should run adflush to clear the cache to ensure your changes take effect.

pam.deny.users

This configuration parameter specifies the users that should be denied access to PAM-enabled applications. When this parameter is defined, only the listed users are denied access. All other users are allowed access.

If you want to use this parameter to control which users can log in, the users you specify should be valid Active Directory users that have been enabled for UNIX. If you specify local user accounts or invalid Active Directory user names, these entries are ignored.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

When a user attempts to log on or access a PAM-enabled service, the pam_ module checks the users specified by this parameter to see if the user is listed there. If the user is included in the list, the user is rejected and authentication fails. If the user is not listed, the user is accepted and authentication proceeds.

The parameter value can be one or more user names, separated by commas or spaces, or the file: keyword and a file location. For example, to prevent the user accounts starr and guestuser fcentrifydcrom logging on:

pam.deny.users: starr,guestuser

You can use the short format of the user name or the full canonical name of the user.

To enter user names with spaces, enclose them in double quotes; for example:

pam.deny.users: "sp1 user@acme.com",joan@acme.com,"sp2 user@acme.com"

To specify a file that contains a list of the users that should be denied access:

pam.deny.users: file:/etc/centrifydc/users.deny

> **Note**: If a computer is configured to use Auto Zone without a zone, enter user names in the format specified by the auto.schema.name.format parameter:

- SAM (samAccountName — this is the default); for example: jcool
- samAccountName@domain_name; for example: jcool@acme.com
- NTLM; for example: acme.com+jcool

> **Note**: You can look in the centrifydc.conf configuration file for the value of auto.schema.name.format, or run adquery user -n to see the UNIX name for any user. For example, to see the UNIX name for jcool (and SAM, the default, is set for auto.schema.name.format), execute the following command, which returns the UNIX name as shown:

```
[root]#adquery user -n jcool
jcool
```

If this parameter is not defined in the configuration file, no user filtering is performed.

> **Note**: If you make changes to this parameter, you should run adflush to clear the cache to ensure your changes take effect.

pam.homedir.create

This configuration parameter specifies whether a new home directory should be created automatically when a new Active Directory user logs on to a system for the first time.

For example, to specify that home directories be created automatically when new Active Directory users log on to a system for the first time:

pam.homedir.create: true

In most cases, you set this configuration parameter using group policy.

> **Note**: For computers that use NFS to mount home directories, you should set this parameter to false. If you have a Solaris environment and set this parameter to true, you should make sure the default location for creating a home directory is not /home/{$user} since this path is not allowed in a typical Solaris environment. In addition, some platforms may require you to manually create a skeleton directory that contains default initial profiles to use when creating new home directories. You can use the pam.homeskel.dir parameter to specify the location of this skeleton directory if it exists in your environment.

pam.homedir.create.hook

By default, the agent uses the mkdir() function to create the user's home directory. However, if desired, you can set this parameter so that you create the user's home directory by way of a script.

The script should be:

- root owned and only writeable by owner
- executable
- not a symlink

When you create the script, be sure that it accepts the following arguments:

- -h (home directory path)
- -u (the UID for the home directory)
- -g (the GID for the home directory)

There is a sample script /usr/share/centrifydc/samples/homedir.sh.sample; this script can create a Solaris ZFS dataset for a user's home directory.

For example:

pam.homedir.create.hook: /usr/bin/homedir.sh

When you specify the script for this parameter, be sure to also specify the absolute path to the script.

pam.homedir.create.mesg

This configuration parameter specifies the message displayed when a user's home directory is created.

For example:

pam.homedir.create.mesg: Created home directory

pam.homedir.perms

This configuration parameter specifies the permissions for a user's home directory if a new home directory is created for the user on the local computer.

For example, to give read, write, and execute permissions on the directory to the user and no other permissions:

pam.homedir.perms: 0700

In most cases, you set this configuration parameter using group policy.

pam.homedir.perms

This configuration parameter specifies the permissions for a user's home directory if a new home directory is created for the user on the local computer.

For example, to give read, write, and execute permissions on the directory to the user and no other permissions:

pam.homedir.perms: 0700

In most cases, you set this configuration parameter using group policy.

pam.homedir.update.ownership

This configuration parameter specifies whether or not to update the home directory ownership when a user logs in.

By default, this parameter is set to false.

```
pam.homedir.update.ownership: true
```

pam.homedir.perms.recursive

This configuration parameter specifies whether to use the permissions defined in the PAM skeleton directory or the permissions defined in pam.homedir.perms when a new home directory is created for a user.

This parameter can have a value of true or false. When set to true, a user's new home directory is created with the contents of the skeleton directory and the permissions defined in pam.homedir.perms. When set to false, a user's new home directory is created using the contents and permissions of the skeleton directory.

This parameter has a default value of false. For example:

pam.homedir.perms.recursive: false

pam.homeskel.dir

This configuration parameter specifies where the PAM skeleton directory is located. The skeleton directory to used to automatically create a new home directory and UNIX profile for a new user, if needed.

The parameter value must be a path name. For example:

pam.homeskel.dir: /etc/skel

If this parameter is not defined in the configuration file, no files are copied when a new user directory is created.

pam.ignore.users

This configuration parameter specifies one or more users that the agent will ignore for lookup in Active Directory. This configuration parameter ignores listed users for authentication and NSS lookups. Because this parameter allows you to intentionally skip looking up an account in Active Directory, it allows faster lookup for system accounts such as tty, root, and bin and local login accounts.

> **Note**: Starting with Delinea DB2 agent 5.2.3, the db2.implement.pam.ignore.users parameter controls whether the agent checks pam.ignore.users. The pam.ignore.users parameter is checked only if db2.implement.pam.ignore.users is set to true. If db2.implement.pam.ignore.users is set to false, pam.ignore.users is not checked, and all users are authenticated in Active Directory. See db2.implement.pam.ignore.users for more information about db2.implement.pam.ignore.users.

In most cases, you set this configuration parameter using group policy. This list is then stored in the /etc/centrifydc/user.ignore file and used to disable lookups in Active Directory for the users specified.You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be one or more user names, separated by a space, or the file: keyword and a file location. For example, to specify a list of users to authenticate locally:

pam.ignore.users: root sys tty

To specify a file that contains a list of the users to ignore:

pam.ignore.users: file:/etc/centrifydc/users.ignore

If this parameter is not defined in the configuration file, no users are specified.

Skipping Active Directory authentication for local AIX users

By default, the agent modifies the AIX Loadable Authentication Module (LAM) for the SYSTEM user attribute to look like this:

SYSTEM=CentrifyDC OR CentrifyDC[NOTFOUND] AND compat

This setting specifies that the first attempt to authenticate a user should be passed to Active Directory through the agent. In some cases, however, you may have local user accounts that you only want to authenticate locally. Although there are parameters in the access control configuration file (centrifydc.conf) that enable you to ignore Active Directory authentication for specific local users, these parameters are not completely applicable on computers running AIX. To exclude any local user account from Active Directory authentication on AIX, you can run the following command for the user:

chuser SYSTEM=compat username

Alternatively, you can edit the /etc/security/user file and change the stanza for a particular user's SYSTEM attribute to:

SYSTEM=compat

If you later decide you want to migrate the local user account to use Active Directory, you can run the following command for the user to reset the default authentication:

chuser SYSTEM= username

> **Note**: To reset the user account to be authenticated through Active Directory, there must be a space after the equal sign (=) in the command line.

pam.mapuser.username

This configuration parameter maps a local UNIX user account to an Active Directory account. Local user mapping allows you to set password policies in Active Directory even when a local UNIX account is used to log in. This parameter is most commonly used to map local system or application service accounts to an Active Directory account and password, but it can be used for any local user account. For more information about mapping local accounts to Active Directory users, see "Mapping local UNIX accounts to Active Directory in the *Administrator's Guide*.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, you should note that the local account name you want to map to Active Directory is specified as the last portion of the configuration parameter name. The parameter value is the Active Directory account name for the specified local user. For example, the following parameter maps the local UNIX account oracle to the Active Directory account oracle_storm@acme.com if the host computer's name is storm:

pam.mapuser.oracle: oracle_$HOSTNAME@acme.com

You can specify the user name in the configuration file with any of the following valid formats:

- Standard Windows format: domain\user_name
- Universal Principal Name (UPN): user_name@domain
- Alternate UPN: alt_user_name@alt_domain
- UNIX user name: user

You must include the domain name in the format if the user account is not in the local computer's current Active Directory domain.

If this parameter is not defined in the configuration file, no local UNIX user accounts are mapped to Active Directory accounts.

pam.mfa.program.ignore

This configuration parameter specifies a list of programs for which multi-factor authentication is ignored. If you have configured roles to require multi-factor authentication, users assigned to those roles will be required to provide two types of authentication to access PAM applications. However, some PAM applications do not support more than one authentication challenge.

You can use this parameter to add the program names that do not support multi-factor authentication. When users access the PAM applications you specify for this parameter, the multi-factor authentication requirement is ignored so that users can log on rather than be denied access.

For example, if you have configured a role with the login-all PAM application right and the Require multi-factor authentication system right, you can use this parameter to skip multifactor authentication for specific PAM applications—such as xscreensaver and vsftpd—where multi-factory authentication is not needed or not supported.

pam.mfa.program.ignore: xscreensaver vsftpd

You can specify multiple options separated by spaces.

By default, ftpd, proftpd, vsftpd, java, httpd, cdc_chkpwd, kdm, and unix2_chkpwd are all added to this parameter.

pam.ntlm.auth.domains

This configuration parameter specifies the list of domains that should use NTLM authentication instead of Kerberos authentication. This parameter enables you to authenticate users behind a firewall when the Kerberos ports are blocked, but a trust relationship exists between domains inside and outside the firewall. When you set this parameter, the local domain controller outside of the firewall passes its authentication requests through the transitive trust chain for authentication inside of the firewall.

The parameter value must be one or more fully-qualified Active Directory domain names. The Active Directory domain names must be mapped to NTLM domain names, either automatically if the firewall does not prevent the mapping from being discovered, or manually by modifying the contents of the /etc/centrifydc/domains.conf file if the firewall prevents the mapping from automatically being discovered.

If firewall constraints prevent the automatic discovery of Active Directory to NTLM domain mapping, you can manually configure how Active Directory domain names map to NTLM domains by editing the /etc/centrifydc/domains.conf file to consist of a list of colon-separated values in the form of:

AD_DomainName:NTLM_DomainName

For example, the domains.conf file should consist of entries similar to the following:

AJAX.ORG:AJAX

FIREFLY.COM:FIREFLY

HR1.FIREFLY.COM:HR1

You can then use the adclient.ntlm.domains parameter using the file: keyword to specify the location of this file. For example:

adclient.ntlm.domains: file:/etc/centrifydc/domains.conf

> **Note**: If you don't want to define the Active Directory to NTLM mapping in a separate file, you can set the adclient.ntlm.domains parameter to map domain names using the format AD_DomainName:NTLM_DomainName. For example:

adclient.ntlm.domains: AJAX.ORG:AJAX FIREFLY.COM:FIREFLY

After you have configured the mapping, you can list the Active Directory domain names for this parameter. For example, to specify that the Active Directory domains AJAX.ORG and FIREFLY.COM, which are outside of the firewall with a one-way trust to the forest inside the firewall, should use NTLM authentication, you could set the parameter like this:

pam.ntlm.auth.domains: AJAX.ORG, FIREFLY.COM

For more information about manually defining the mapping of Active Directory domains to NTLM domains, see adclient.ntlm.domains.

Alternatively, you can set the group policy **Computer Configuration > Centrify Settings > DirectControl Settings > Pam Settings > Specify NTLM authentication domains**.

pam.password.change.mesg

This configuration parameter specifies the text displayed by a PAM-enabled application when it requests a user to change a password.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an ASCII string. UNIX special characters and environment variables are allowed. For example:

pam.password.change.mesg: Changing Active Directory password for\

If this parameter is not present, its default value is “Change password for”.

pam.password.change.required.mesg

This configuration parameter specifies the message displayed if the user enters the correct password, but the password must be changed immediately.

For example:

pam.password.change.required.mesg: \
You are required to change your password immediately

pam.password.confirm.mesg

This configuration parameter specifies the text displayed by a PAM-enabled application when it requests a user to confirm his new password by entering it again.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an ASCII string. UNIX special characters and environment variables are allowed. For example:

pam.password.confirm.mesg: Confirm new Active Directory password:\

If this parameter is not present, its default value is “Confirm new password:”.

pam.password.empty.mesg

This configuration parameter specifies the message displayed if the user enters an empty password.

For example:

pam.password.empty.mesg: Empty password not allowed

pam.password.enter.mesg

This configuration parameter specifies the text displayed by a PAM-enabled application when it requests a user to enter his password.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an ASCII string. UNIX special characters and environment variables are allowed. For example:

pam.password.enter.mesg: Active Directory password:\

If this parameter is not present, its default value is “Password:”.

pam.password.expiry.warn

This configuration parameter specifies how many days before a password expires the PAM-enabled applications should start issuing the pam.password.expiry.warn.mesg to the user.

The parameter value must be a positive integer. For example, to issue a password expiration warning 10 days before a password is set to expire:

pam.password.expiry.warn: 10

If this parameter is not present, the default value is 14 days.

pam.password.expiry.warn.mesg

This configuration parameter specifies the text displayed by a PAM-enabled application to warn the user that her password will expire in pam.password.expiry.warn days or less.

When the message is displayed, the '%d' token is replaced with the number of days until expiration. The message must contain exactly one '%d' token and no other '%' characters.

For example:

pam.password.expiry.warn.mesg: Password will expire in %d days

pam.password.new.mesg

This configuration parameter specifies the text displayed by a PAM-enabled application when it requests a user to enter his new password during a password change.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an ASCII string. UNIX special characters and environment variables are allowed. For example:

pam.password.new.mesg: Enter new Active Directory password:\

If this parameter is not present, its default value is “Enter new password:”.

pam.password.new.mismatch.mesg

This configuration parameter specifies the message displayed during password change when the two new passwords do not match each other.

For example:

pam.password.new.mismatch.mesg: New passwords don't match

pam.password.old.mesg

This configuration parameter specifies the message displayed by a PAM-enabled application when it requests a user to enter his old password during a password change.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The parameter value must be an ASCII string. UNIX special characters and environment variables are allowed. For example:

pam.password.old.mesg: (current) Active Directory password:\

If this parameter is not present, its default value is "(current) password:".

pam.policy.violation.mesg

This configuration parameter specifies the message displayed during password change if the operation fails because of a domain password policy violation. For example, if the user attempts to enter a password that doesn't contain the minimum number of characters or doesn't meet complexity requirements, this message is displayed.

For example:

pam.policy.violation.mesg: \
The password change operation failed due to a policy restriction set by the Active Directory administrator.
This may be due to the new password length, lack of complexity or a minimum age for the current password.

pam.setcred.respect.sufficient

This configuration parameter overrides an anomaly in the operation of the PAM interface on some platforms that denies access to a user who has entered the correct password. The default setting depends upon the platform as follows:

- For HPUX and Mac OSX platforms the default is true
- For all other platforms the default is false.

  Note: Some Solaris 2.6 and Solaris 8 users have reported getting the error message PAM_AUTH_ERR after entering the correct password. If this occurs, set pam.setcred.respect.sufficient: true.

pam.setcred.support.refresh

This parameter specifies whether the PAM flag PAM_REFRESH_CRED is supported and can be used to trigger creation of the credential cache and renew Kerberos tickets. The default is false, in which case the PAM_ESTABLISH_CRED flag is used to trigger creation of the credential cache and renew Kerberos tickets. For example:

pam.setcred.support.refresh: false

pam.setcred.support.reinitialize

This parameter specifies whether the PAM flag PAM_REINITIALIZE_CRED is supported and can be used to trigger creation of the credential cache and renew Kerberos tickets. The default is false, in which case the PAM_ESTABLISH_CRED flag is used to trigger creation of the credential cache and renew Kerberos tickets.

For example:

pam.setcred.support.reinitialize: false

pam.sync.mapuser

This configuration parameter controls whether the password synchronization service keeps passwords synchronized for local users that are mapped to an Active Directory account.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you set this parameter in the configuration file, the parameter value should be a list of local user accounts that are mapped to Active Directory accounts. For example:

pam.sync.mapuser: root oracle sanchez

If you set this parameter and a mapped user changes his password, PAM updates the password hash for the corresponding local UNIX account in the local /etc/shadow file so that the passwords match. Synchronizing the passwords in this way ensures that local users can still log on even if there are problems with the network, Active Directory, or the adclient process. For example, if Active Directory is not available, the mapped user can log on as a local user by appending @localhost to the user name:

sanchez@localhost

Password synchronization requires you to do the following:

- Install either the Delinea Password Synchronization component or the Microsoft Password Synchronization Service on all domain controllers.

  If you do not have the Microsoft Password Synchronization Service installed on your domain controllers, you can install and use the Delinea Password Synchronization extension instead. You can install the Delinea Password Synchronization extension when you install other Delinea Management Services using the setup program or by running the standalone password extension installation program.

- Configure the zone properties for the computer's zone to support agentless clients and to use the proper NIS domain name and Active Directory attribute for storing the user's password hash.

- Map the specified local users to Active Directory using either the pam.mapuser.username configuration parameter or group policy.

- Verify the Active Directory user to which the local user is mapped has a profile in the zone you have configured for agentless authentication.

This parameter has no effect on Mac OS X systems.

pam.uid.conflict

This configuration parameter specifies how you want the agent to respond if a user logs on with an Active Directory account and either the Active Directory user name or Active Directory UID conflicts with a local user account. The purpose of detecting a duplicate user name or duplicate UID is to prevent an Active Directory user from signing on and receiving privileges to modify files created by a different local user.

The pam.uid.conflict configuration parameter determines what happens when this type of conflict is found. The parameter value must be set to one of the following valid options:

ignore	Do not report duplicate user names or UID conflicts. If detected, log the conflict at the info level if logging is enabled.
warn	Warn the user of the user name or UID conflict after s successful login. Log the conflict at warning level if logging is enabled. This is the default value.
error	Report UID conflict to user after user name is entered. Don't accept password. Don't allow log in. Log conflict at error level.

For example:

pam.uid.conflict: warn

> **Note**: If both the Active Directory user name and Active Directory UID are the same as a local user name and UID, the accounts do not conflict and the user can log on regardless of how you set this parameter. Although this situation is rare, you should avoid using Active Directory user names and UIDs that duplicate local user names and UIDs but apply to different individual users.

If this parameter is not present, its default value is warn.

pam.workstation.denied.mesg

This configuration parameter specifies the message displayed if a user attempting to log on is denied access because of a workstation restriction.

For example:

pam.workstation.denied.mesg: \
Your account is configured to prevent you from using this computer. Please try another computer.

microsoft.pam.privilege.escalation.enabled

The configuration parameter specifies if the Microsoft Privileged Access Management (PAM) Privilege Escalation feature is supported or not within the Delinea environment.

If microsoft.pam.privilege.escalation.enabled is true, then, when an Active Directory user logs in, the configured privilege that's granted to the user through PAMGroup takes effect until the granted period has elapsed.

The Privileged Access Management (PAM) Privilege Escalation feature can be enabled or disabled through Group Policy. Select **Computer Configuration › Centrify Settings › DirectControl Settings › Enable Active Directory PAM Privilege Escalation feature**

The Microsoft PAM Privilege Escalation feature specifies if Delinea DirectControl uses Microsoft PAM Privilege Escalation feature in the computer.

For example:

microsoft.pam.privilege.escalation.enabled: true

Default is false, the Microsoft PAM Privilege Escalation feature support is disabled. Setting it to true enables grants the Active Directory user, at log in, the same configured privilege as the user's PAMGroup. This is in effect until the grant period expires.

Customizing Group Policy Configuration Parameters

This section describes the configuration parameters that affect group policy support on the local host computer.

gp.disable.all

This configuration parameter can be used to disable both computer and user group policies on a local computer. If set to true, all group policy settings are ignored.

For example:

gp.disable.all: true

If this parameter is not defined in the configuration file, its default value is false.

gp.disable.machine

This configuration parameter can be used to disable computer-based group policies on a local computer. If set to true, all computer-based group policy settings are ignored.

For example:

gp.disable.machine: true

If this parameter is not defined in the configuration file, its default value is false.

gp.disable.user

This configuration parameter can be used to disable user-based group policies on a local computer. If set to true, all user-based group policy settings are ignored.

For example:

gp.disable.user: true

If this parameter is not defined in the configuration file, its default value is false.

gp.disk.space.check.folders

This configuration parameter specifies the folders that need the free disk space check. If the free space in any specified folder is less than the value in gp.disk.space.min, then group policy settings will not be updated.

Specify a comma-separated list of folders; for example, the default is:

gp.disk.space.check.folders: /,/etc,/var

Delinea

gp.disk.space.min

This configuration parameter specifies the minimum free disk space in kilobytes (KB) that is required for a group policy update. If the free disk space in any folder specified in gp.disk.space.check.folders is less than this value, then group policy settings will not be updated.

When updating the configuration file, the Perl mapper scripts create a temp file, print to it, and replace the original file. If the disk is full, the mapper cannot write to the temp file, so the temp file is empty, and the original file is replaced by the empty temp file. This configuration parameter and gp.disk.space.min prevent the mapper writing to a temp file when disk space is low.

The default value is 5120 KBytes. Set this parameter to 0 to not check free disk space.

gp.mappers.certgp.pl.additional.cafiles

This setting defines a list of certificate files which will be included in the certgp.pl install, if found.

It can be a list of certificates to be added. For example:

gp.mappers.certgp.pl.additional.cafiles: <ca-file> <ca-file> ...

It can also point to a file that contains a list of certificate files to be added. For example:

gp.mappers.certgp.pl.additional.cafiles: file:/etc/centrifydc/cert_included.list

The default value is empty.

gp.mappers.certgp.pl.exclude.cacerts

This setting defines a certificate list which will be excluded from the certgp.pl install, if matched.

It can be a list of fingerprints of the certificates to be excluded. For example:

gp.mappers.certgp.pl.exclude.cacerts: <fingerprint> <fingerprint> ...

It can also point to a file that contains a list of fingerprints of the certificates to be excluded. For example:

gp.mappers.certgp.pl.exclude.cacerts: file:/etc/centrifydc/cert_excluded.list

The default value is empty.

gp.mappers.directory.machine

This configuration parameter specifies the root directory that contains all of the mapping programs for computer-based group policy settings. Individual programs map entries from the virtual registry into configuration settings in the appropriate files on the local computer.

The parameter value must be a path name. For example:

gp.mappers.directory.machine: /usr/share/centrifydc/mappers/machine

If this parameter is not defined in the configuration file, its default value is /usr/share/centrifydc/mappers/machine.

gp.mappers.directory.user

This configuration parameter specifies the root directory that contains all of the mapping programs for user-based group policy settings. Individual programs map entries from the virtual registry into configuration settings in the appropriate files on the local computer.

The parameter value must be a path name. For example:

gp.mappers.directory.machine: /usr/share/centrifydc/mappers/user

If this parameter is not defined in the configuration file, its default value is /usr/share/centrifydc/mappers/user.

gp.mappers.error_file

This configuration parameter specifies the name of the file where the group policy mapper programs write error messages.

For example:

gp.mappers.error_file: mapper.errors

gp.mappers.machine

This configuration parameter specifies the list of mapping programs to run to configure computer-based policies. The mapping programs are contained in the root directory specified by gp.mappers.directory.machine (/usr/share/centrifydc/mappers/machine by default). The mapping programs are executed in the order in which they are specified. The mapping program centrifydc.conf.pl will always run even if unspecified and does not run only if you specify that it not run (described later).

In most cases you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you want to temporarily override group policy.

To specify mapping programs to run, you can list each individual program name literally, or you can use wild card characters that are a subset of regular expression wild card characters:

- An asterisk (*) specifies any set of zero or more characters. "map*", for example, specifies any program names starting with "map". "set*.pl" specifies any program names starting with "set" and ending with ".pl". And "*dc*" specifies any program names that include "dc". "*" means all programs.
- A question mark (?) specifies any single character. "map???", for example, specifies any six-character program name starting with "map".
- Square brackets ([ ]) enclosing a set of characters specifies a single character that is one of the enclosed characters. "mapprogram[123]", for example, matches the program names mapprogram1, mapprogram2, and mapprogram3.

You can specify a program name *not* to execute by preceding it with an exclamation point (!). If you specify "* !mapprogram1", for example, you specify that all mapping programs in the mapping program root directory should execute except for "mapprogram1". Note that the only way you can stop the automatically executing program centrifydc.conf.pl from executing is to specify "!centrifydc.conf.pl" in this parameter.

You can combine all of these rules to give you precise control over which mapping programs run. Some examples:

gp.mappers.machine: * specifies all mapping programs in the mapping program parent directory.

gp.mappers.machine: mapgp* !mapgp2 specifies all mapping programs in the mapping program parent directory that start with "mapgp" *except* for "mapgp2". Note that centrifydc.conf.pl will execute because it hasn't been specified not to execute and so executes automatically.

gp.mappers.runcommand.as.root.env.list

This configuration parameter specifies the list of environment variables that are exported to the environment so that a root user can run Group Policy commands. Use the forward slash "/" to specify environment variables and their values. Use a space to separate multiple name/value pairs. If an environment variable contains spaces, enclose the value in quotes or add a backslash "\" before the space.

You can also specify other environment variables as a value.

For example:

gp.mappers.runcommand.as.root.env.list: env1/value1 env2/va\ lue2 env3/"va lue3" env4/value4_$

> **Note**: If the gp.mappers.runcommand.as.user parameter is set to true, the gp.mappers.runcommand.as.root.env.list has no effect for user Group Policy commands because the service runs the user's group policy commands as the user's login shell and resets all environment variables.

gp.mappers.runcommand.as.user

This configuration parameter specifies whether to run user group policy commands as the current user. The default for this parameter is false, which means that the service runs these commands as root.

The RunCommand.pl script reads this parameter setting and if the parameter is set to true, then it runs commands as follows:

su - $args ->user() -c $command

gp.mappers.runmappers

This configuration parameter specifies the location of the runmappers program. The runmappers program is started by the agent and invokes individual mapping programs for computers, users or both.

The parameter value must be a path name. For example:

gp.mappers.runmappers: /usr/share/centrifydc/mappers/runmappers

If this parameter is not defined in the configuration file, its default value is /usr/share/centrifydc/mappers/runmappers.

gp.mappers.timeout

This configuration parameter specifies the maximum time, in seconds, to allow for a single mapping program to complete execution. If a mapping program takes longer than this period to successfully complete its execution, the process is stopped and the next mapping program is started.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer that is less than the value set for the gp.mappers.timeout.all parameter. For example, to set the timeout interval to 60 seconds:

gp.mappers.timeout: 60

The default value for this parameter on Mac is 120 seconds.

The default value for this parameter on all other platforms is 30 seconds.

gp.mappers.timeout.all

This configuration parameter specifies the maximum time, in seconds, to allow for all mapping programs to complete execution. The parameter value must be a positive integer that is less than the value set for the lrpc.timeout parameter.

The default value for this parameter on Mac is ten minutes (600 seconds). For example:

gp.mappers.timeout.all: 600

The default value for this parameter on all other platforms is four minutes (240 seconds). For example:

gp.mappers.timeout.all: 240

gp.mappers.umask

This configuration parameter specifies the default umask for mapping programs that create files. The default value for this parameter sets the following read and write permissions for mapping programs that create files:

u=rwx
g=rx
o=

The parameter value specifies these permissions using numeric mode. For example:

gp.mappers.umask: 0027

gp.mappers.user

This configuration parameter specifies the mapping programs that map user-based policy settings to run. The mapping programs are executed in the order in which they are specified.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you want to temporarily override group policy.

In defining the list of mapping programs to run, you can use an asterisk (*) as a wild card to match a set of program names. For example, you can specify a* to match all programs with names that start with the letter a. You can use square brackets ([ ]) to match any character within the brackets. For example, you can specify mapprogram[123] to match the program names of mapprogram1, mapprogram2, and mapprogram3. You can also use an exclamation point (!) with a program name to exclude a program from the list. For example, you can specify !mysample to prevent the mapping program mysample from running.

To run all of the mapping programs for user-based policy settings, you can specify:

gp.mappers.user: *

To run a subset of the mapping program, you can explicitly define the order and which programs to run. For example, to run the program mapgp1, followed by mapgp4 and mapgp3, but skipping the execution of mapgp2:

gp.mappers.user: mapgp1 !mapgp2 mapgp4 mapgp3

gp.refresh.disable

This configuration parameter specifies whether to disable the background processing of group policy updates. This configuration parameter applies to both computer- and user-based policies.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you want to temporarily override group policy. For example:

gp.refresh.disable: false

gp.reg.directory.machine

This configuration parameter specifies the root directory of the virtual registry for computer-based group policies. The parameter value must be a path name. For example:

gp.reg.directory.machine: /var/centrifydc/reg/machine

If this parameter is not defined in the configuration file, its default value is/var/centrifydc/reg/machine.

gp.reg.directory.user

This configuration parameter specifies the root directory of the virtual registry for user-based group policies.

The parameter value must be a path name. For example:

gp.reg.directory.user: /var/centrifydc/reg/users

If this parameter is not defined in the configuration file, its default value is /var/centrifydc/reg/users.

gp.use.user.credential.for.user.policy

This configuration parameter specifies whether to use the user's credentials to retrieve user group policies. By default, all group policies are retrieved using the computer account credentials, which are associated with the adclient process rather than the user who is currently logged on. In most cases, this default behavior is sufficient because most of the Delinea group policies are computer configuration policies. However, if the computer account does not have permission to access the Group Policy Object where user policies are defined, the default behavior prevents user policies from being applied.

You can set this configuration parameter to true to use the user's credentials to retrieve user group policies. For example:

gp.use.user.credential.for.user.policy: true

If this parameter is not defined in the configuration file, its default value is false.

gp.user.login.run

This configuration parameter specifies when user-based group policies should run. By default, user-based group policies are applied when a user first logs on to a computer, then at a regular interval in background to check for updates and changes while the user's session remains active. However, running group policies at every login and refresh interval for users who are already logged on can impact performance on computers where there are a large number of group policies being applied. You can use this parameter to reduce the load on those computers by customizing when group policies should be applied.

This configuration parameter enables you to specify whether the user-based group policies should be applied:

- Only once when the user first logs on and not again until the user logs off and logs back on.
- When the user first logs on and regularly at the refresh interval for as long as the user remains logged on.
- Never when the user logs on, but periodically at the refresh interval thereafter.

The valid parameter values for this configuration parameter are once, always, and never.

For example, to specify that user-based group policies should only run once when the user first logs on but not thereafter, you can set this parameter to once:

gp.user.login.run: once

If this parameter is not defined in the configuration file, its default value is always to apply the user group policies when a user first logs on and periodically refresh the policies in the background for as long the user remains logged on.

Customizing NSS-Related Configuration Parameters

This section describes the configuration parameters that affect the operation of NSS-related activity on the local host computer.

> **Note**: On AIX, the NSS configuration parameters described in this chapter may apply to interfaces in the AIX Loadable Authentication Module (LAM). For consistency across platforms, most of the parameter names are the same and retain the reference to NSS settings they configure, but NSS is not used on AIX.

nss.alias.source

This configuration parameter specifies the source to look up aliases, and you specify one of the following values:

- nismaps (default)
- mail
- proxyaddresses

To look up the alias from an Active Directory user object, use the mail or proxyAddresses value. Because proxyaddresses is a custom attribute, you need to also include it in the adclient.custom.attributes.user parameter or else the alias source reverts to nismaps.

Using the mail or proxyAddresses values don't work with users in a one-way trusted forest.

nss.gecos.attribute

This configuration parameter specifies the Active Directory user object attribute to use for the GECOS field. The default value for this parameters is the gecos attribute in the Active Directory RFC2307 schema.

The order of precedence for the GECOS field setting is:

1. The GECOS setting for the UNIX service connection point (SCP) in Active Directory.
2. The nss.gecos.attribute setting.
3. The displayName attribute of the user object.

If nss.gecos.attribute is set and GECOS is not set for the UNIX SCP, the user attribute specified by nss.gecos.attribute is used for the GECOS field in UNIX profiles and NSS lookups. If nss.gecos.attribute is not defined or the Active Directory RFC2307 schema is not used, the user object's displayName attribute is used as the GECOS field for UNIX profiles.

If you set this configuration parameter, the parameter value is case-sensitive and must exactly match the case used for the attribute name in Active Directory. For example:

nss.gecos.attribute: displayName

nss.gid.ignore

This configuration parameter specifies a set of one or more group identifiers that the Delinea NSS module will ignore for lookup in Active Directory.

In most cases, this configuration parameter's value is generated automatically by group policy.

If you select the **Specify group names to ignore** policy and click **Enabled**, you can type the list of local group names not stored in Active Directory. The list you specify for the group policy is then stored in the /etc/centrifydc/group.ignore file and used to automatically generate the /etc/centrifydc/gid.ignore file. These files are then used to disable looking up account information in Active Directory for the groups specified, which results in faster name lookup service for system group accounts such as tty and disk.

You can, however, define this parameter manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you manually set this parameter, the parameter value should be one or more group identifiers, separated by a space, or the file: keyword and a file location. For example:

nss.gid.ignore: 0 20 5861
nss.gid.ignore=file:/etc/centrifydc/gid.ignore

A default set of groups to ignore are defined in sample /etc/centrifydc/group.ignore and /etc/centrifydc/gid.ignore files. If you edit either file, be sure to run the adreload command after modifying the file to have the changes take effect.

nss.group.ignore

This configuration parameter specifies a set of one or more groups that the Delinea NSS module will ignore for lookup in Active Directory.

In most cases, this configuration parameter's value is generated automatically by group policy.

If you select the **Specify group names to ignore** policy and click **Enabled**, you can type the list of local group names not stored in Active Directory. The list you specify for the group policy is then stored in the /etc/centrifydc/group.ignore file and used to automatically generate the /etc/centrifydc/gid.ignore file. These files are then used to disable looking up account information in Active Directory for the groups specified, which results in faster name lookup service for system group accounts such as tty and disk.

You can, however, set this parameter manually in the configuration file if you aren't using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be one or more group names, separated by a space, or the file: keyword and a file location. For example:

nss.group.ignore: maintenance apps
nss.group.ignore=file:/etc/centrifydc/group.ignore

A default set of groups to ignore are defined in sample /etc/centrifydc/group.ignore and /etc/centrifydc/gid.ignore files. If you are not using group policies, you can uncomment the nss.group.ignore parameter in the /etc/centrifydc/centrifydc.conf file to ignore the default set of groups.

> **Note**: If you plan to edit the group.ignore file, be sure to run the adreload command after modifying the file to have the changes take effect.

nss.group.override

This configuration parameter allows you to override group profile entries for zone groups. Using this parameter is similar to defining override filters for local groups in the /etc/group file. By defining override filters, you can use this parameter to give you fine-grain control over the groups that can access a local computer. You can also use the override controls to modify the information for specific fields in each group entry on the local computer. For example, you can override the group ID or member list for a specific group on the local computer without modifying the group entry itself.

In most cases, you set this configuration parameter using group policy. The entries created by group policy are then stored in the /etc/centrifydc/group.ovr file and used to filter group access to a local computer.You can, however, set this parameter manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The syntax for overriding group entries is similar to the syntax used for overriding NIS. You use + and – entries to allow or deny access for specific groups on the local system. Additional fields correspond to the standard /etc/group fields separated by colons (:).

In most cases, the nss.group.override parameter is used to identify a file location of an override file that contains all of group override entries you want to use on the local computer. For example:

nss.group.override: file:/etc/centrifydc/group.ovr

Within the override file, you use the following format:

+zone_group_name:group_name:group_password:group_id:member_list
-zone_group_name:group_name:group_password:group_id:member_list

For example:

+users::::
+adminsjdoe,bsmith,frank
+ftpusers:ftp::300:
-webusers
+

> **Note**: Changes to the group password field are ignored.

For more information about overriding group entries, see the sample group override file /etc/centrifydc/group.ovr.

> **Note**: If you make changes to this parameter or the override file, you should run adflush to clear the cache to ensure your changes take effect.

nss.group.skip.members

This configuration parameter allows you to skip the retrieval of group membership information for specific groups. Retrieving group membership information from Active Directory can be a very time-consuming and memory-intensive operation for groups with a large number of users, or when using nested groups, but in many cases this information is not needed to perform common UNIX operations. Using this configuration parameter to skip the retrieval of group membership information for specific groups can greatly improve performance for groups with a large number of members.

The parameter value should be a comma-separated list of the UNIX commands for which you can skip group member expansion in the getgrent() call.

The default setting for this configuration parameter is the following for most systems:

ls,chown,find,ps,chgrp,dtaction,dtwm,pt_chmod,id,login,sshd,sshd2,getty,dtlogin,su,adsetgrps,adid

For AIX system, the default is the following:

nss.group.skip.members=ls,chown,find,ps,chgrp,dtaction,dtwm,pt_chmod,id, login,sshd,sshd2,getty,dtlogin,su,adsetgrps,adid

> **Note**: Setting this parameter does not affect the information returned when the nscd or pwgrd daemon is running on a system. The nscd or pwgrd daemons provide a cache for faster user and group lookups, but when the response comes from this cache, the agent cannot modify the response to skip the members listed with this parameter.

nss.nobody.gid

This configuration parameter specifies the group ID (GID) of the system's nobody group.

For example:

nss.nobody.gid: 99

nss.nobody.group

This configuration parameter specifies the group name of the system's nobody group.

For example:

nss.nobody.group: nobody

nss.nobody.uid

This configuration parameter specifies the user ID (UID) of the system's nobody user.

For example:

nss.nobody.uid: 99

nss.nobody.user

This configuration parameter specifies the user name of the system's nobody user.

For example:

nss.nobody.user: nobody

nss.passwd.hash

This configuration parameter specifies whether to include the UNIX password hash in response to the getpw* commands. The parameter value can be true or false. The default value for the parameter is false because the password hash is sensitive information and can make a system vulnerable to a brute force attack. However, if you have applications, such as Informix, that validate users based on the password hash retrieved from NSS, you can set this parameter to true to accommodate those applications.

If you set this parameter to true, however, you must also install a password synchronization service on all of the domain controllers in the domain. The password synchronization service can be the Delinea Password Filter, or the Password Synchronization Service provided by Microsoft in Windows Server 2003 R2 or in the Microsoft Services for UNIX (SFU) package.

nss.passwd.info.hide

This configuration parameter specifies whether to hide the following password attributes from non-root users:

- Maximum Password Age
- Password Expiration Date
- Minimum Password Age
- Change Password Needed
- Password Last Changed On

The parameter value can be true or false. When this parameter is set to true, only users with root permissions can view the password attributes shown above. When this parameter is set to false, users without root permissions can view the password attributes.

The default value for this parameter is true on all UNIX operating systems except HP-UX. On HP-UX, the default is false because HP-UX does not support hiding these attributes.

nss.passwd.override

This configuration parameter allows you to override user profile entries for zone users. Using this parameter is similar to defining override filters for local users in the /etc/passwd file. By defining override filters, you can use this parameter to give you fine-grain control over the user accounts that can access a local computer. You can also use the override controls to modify the information for specific fields in each /etc/passwd entry on the local computer. For example, you can override the user ID, primary group ID, default shell, or home directory for specific login accounts on the local computer without modifying the account entry itself.

In most cases, you set this configuration parameter usinggroup policy. The entries created by group policy are then stored in the /etc/centrifydc/passwd.ovr file and used to filter user access to a local computer.You can, however, set this parameter manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The syntax for overriding passwd entries is similar to the syntax used for overriding NIS. You use + and – entries to allow or deny access for specific users on the local system. Additional fields correspond to the standard /etc/passwd fields separated by colons (:).

In most cases, the nss.passwd.override parameter is used to identify a file location of an override file that contains all of passwd override entries you want to use on the local computer. For example:

nss.group.override: file:/etc/centrifydc/passwd.ovr

> **Note**: Although the passwd.ovr file is generated from the list of override entries you specify using group policy, you can also manually create or update the override file on any local computer, if needed. A sample illustrating the syntax is provided in the /etc/centrifydc/passwd.ovr.sample file.

Within the override file, you use the following format for entries:

+zone_username:username:password:uid:gid:GECOS:home_directory:shell

For example:

```
+mike:::::::/usr/local/ultrabash
+kris:kdavis:x:6:6:Kris Davis:/home/kdavis:/bin/bash
+janedoe@acme.test:jdoe300:300:
+@sysadmins:::::::
-ftp
+@staff:::::::
+@rejected-users:32767:32767:/bin/false
+:/sbin/nologin
+:
```

> **Note**: Overriding the password hash field is ignored. Changing this field in the override file does not affect zone user passwords. In overriding passwd entries, users accounts must be enabled for UNIX in the zone, but the groups do not need to be UNIX-enabled.

In the example above, the @ symbol denotes an Active Directory name. It may be an Active Directory group name, a zone name, or some other container name. You may also specify an Active Directory user principal name instead of the zone name.

Entries in the override file are evaluated in order from first to last with the first match taking precedence. This means the system will only use the first entry that matches a particular user. For example, if the user cruz is a member of both the staff group and the rejected-users group and you have defined the override entries as listed in the example above, the cruz user account is allowed to log on to the computer because the staff entry is evaluated and matched before the rejected-users entry. If the order were reversed in the override file, the cruz account would be flagged as a rejected-users account and denied access.

> **Note**: If you manually create the passwd.ovr file, you must include the following as the last line in the file:

+::::::::

For more information about overriding group entries, see the sample passwd override file /etc/centrifydc/passwd.ovr. For information about using the NSS Overrides group policy to generate and maintain the passwd.ovr file, see the Access Manager online help.

> **Note**: If you make changes to this parameter or the override file, you should run adflush to clear the cache to ensure your changes take effect.

nss.process_group.ignore

This configuration parameter specifies a set of one or more process groups that the Delinea NSS module will ignore for lookup in Active Directory. A *process group* is a collection of related processes which can be called at the same time.

nss.program.ignore

This configuration parameter specifies one or more programs that should not look up account information in Active Directory. The programs you specify for this parameter do not use the agent to contact Active Directory.

Setting this parameter helps to ensure that local programs that create, manage, or use local user and group information do not attempt to look up conflicting information in Active Directory. For example, you can specify programs such as adduser and addgroup to ensure those programs can still be used to create and update local accounts independent of Active Directory:

nss.program.ignore: addgroup,adduser

The specific programs you should include in the list vary by platform and the specific operating environment you are using. The default setting for this configuration parameter includes the most common program names that shouldn't make calls to Active Directory through the agent.

If you have auditing enabled, the agent's auditing service maintains a cache of user information for performance reasons. When you have auditing enabled, you can also use this parameter to circumvent the agent accessing its local cache when you use commands that manipulate local user information directly. For example, you would want the agent to skip checking its local cache when you use commands such as useradd, userdel, adduser, usermod, mkuser, rmuser, chuser, and any other programs that directly access the local /etc/passwd file.

> **Note**: Setting this parameter does not affect the information returned when the nscd or pwgrd daemon is running on a system. The nscd and pwgrd daemons provide a cache for faster user and group lookups, but when the response comes from this cache, the agent cannot modify the response to skip the programs listed with this parameter.

You can also set this configuration parameter usinggroup policy.

nss.program.ignore.check.parents

This configuration parameter specifies whether to recursively check the parent processes' names for the "nss.program.ignore" parameter. The default value is false.

You can also set this configuration parameter using group policy.

nss.shell.emergency.enabled

This configuration parameter specifies whether to use the default login shell when a user or group attempting to access the computer is not allowed to log in.

The default no-login shell and its location is typically platform-specific. For example, on machines running Red Hat Linux, the default shell for users who are denied access is:

/sbin/nologin

The default for this parameter is false, which means that the nologin shell configured in nss.shell.nologin is returned.

nss.shell.nologin

This configuration parameter specifies the default login shell to use when a user or group attempting to access the computer is not allowed to log on. The default no-login shell and its location is typically platform-specific. For example, on Red Hat Linux the default shell for users who are denied access is /sbin/nologin.

For example:

nss.shell.nologin: /sbin/nologin

> **Note**: If you make changes to this parameter, you should run adflush to clear the cache to ensure your changes take effect.

nss.split.group.membership

This configuration parameter specifies whether to split up or truncate large groups when you use the getent group UNIX command to retrieve group information.

In operating environments that do not support large groups, commands that return group information could fail or return incomplete results when a group has a membership list exceeds the maximum size allowed. Typically, the maximum size allowed for groups is 1024 bytes, which is roughly equivalent to 125 users. If your environment contains large groups that exceed the 1024-byte limit, you can set this parameter to true to have those groups automatically split into multiple groups when they reach the maximum size.

When this parameter is set to true and you issue the getent group command without specifying a group name, large groups are split into sublists, and all sublists are returned. When this parameter is set to false, large groups are truncated, and only the truncated results of the group list (typically the first 1024 bytes) are returned.

> **Note**: This policy has no effect in Mac OS X environments.
>
> **Note**: This configuration parameter takes effect only when you do not specify a group name on the getent group command line. Because of the way in which group information is queried in NSS, group lists are always truncated (and not split) when you specify a group name on the getent group command line (for example, getent group group_name).

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

The default value is true for Solaris, HPUX, and IRIX, but false for all other operating environments. For example:

nss.split.group.membership: true

nss.squash.root

This configuration parameter specifies whether you want to force root and wheel super-user accounts to be defined locally. If you set this parameter to true, Active Directory users with a UID of 0, a GID of 0, a user or group name of root, or a group name of wheel are not permitted to log on. Because the agent cannot prevent Active Directory users or groups from being assigned a UID or GID of 0, which would give those users or groups root-level access to the computers in a zone, you can use this parameter to prevent any Active Directory users with a UID or GID of 0 from logging on. Setting this parameter to true forces the privileged accounts to be defined as local accounts and not authenticated through Active Directory.

For example:

nss.squash.root: true

If you set this parameter to false, you should use other configuration parameters, such as pam.ignore.users or user.ignore to skip Active Directory authentication for system accounts so that Active Directory users cannot be granted root access on the computers in the zones they are permitted to access.

The default value for this parameter is true. It is possible, however, for an Active Directory administrator to override this setting through the use of group policy applied to a local computer, for example, by using the **Sudo rights** group policy. There is no way to effectively prevent the setting from being changed, except by disabling computer-based group policies in the local centrifydc.conf file or by strictly controlling who has permission to enable and apply group policies to computers that join an Active Directory domain. For information about disabling group policies using parameters in the local centrifydc.conf file, see gp.disable.all or gp.disable.machine in Customizing

nss.uid.ignore

This configuration parameter specifies a set of one or more user identifiers that the Delinea NSS module will ignore for lookup in Active Directory.

In most cases, this configuration parameter's value is generated automatically by group policy.

If you select the **Specify user names to ignore** group policy and click **Enabled**, you can type the list of local user names not stored in Active Directory. The list you specify for the group policy is then stored in the /etc/centrifydc/user.ignore file and used to automatically generate the /etc/centrifydc/uid.ignore file. These files are then used to disable looking up account information in Active Directory for the users specified, which results in faster name lookup service for system user accounts such as tty and disk.

You can, however, define this parameter manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you manually set this parameter, the parameter value should be one or more user identifiers, separated by a space, or the file: keyword and a file location. For example:

nss.uid.ignore: 0 20 5861
nss.uid.ignore=file:/etc/centrifydc/uid.ignore

A default set of system user accounts to ignore is defined in the sample /etc/centrifydc/user.ignore file and in the /etc/centrifydc/uid.ignore file. If you edit either file, be sure to run the adreload command after modifying the file to have the changes take effect.

nss.user.group.prefer.cache

Use this parameter to specify whether or not to always use the cached information and defer the refreshing of Active Directory information in the background.

If you have a lot of NSS queries for users or groups, you can improve adclient performance by enabling this parameter.

The default value for this parameter is false.

This parameter applies to NSS user and group queries (getpw* and getgr*).

nss.user.ignore

This configuration parameter specifies one or more users that the Delinea NSS module will ignore for lookup in Active Directory. Because this parameter allows you to intentionally skip looking up specific accounts in Active Directory, it allows faster lookup for system accounts such as tty, root, and bin.

> **Note**: This configuration parameter only ignores the listed users for NSS lookups. To ignore users for authentication and NSS lookups, use the pam.ignore.users configuration parameter.

In most cases, this configuration parameter's value is generated automatically by group policy.

If you select the **Specify user names to ignore** policy and click **Enabled**, you can type the list of local user names not stored in Active Directory. This list is then stored in the /etc/centrifydc/user.ignore file and used to automatically generate the /etc/centrifydc/uid.ignore file. These files are then used to disable looking up account information in Active Directory for the users specified, which results in faster name lookup service for system user accounts such as tty and disk.

You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value should be one or more user names, separated by a space, or the file: keyword and a file location. For example:

```
nss.user.ignore: root sys tty
nss.user.ignore=file:/etc/centrifydc/user.ignore
```

A default set of users to ignore are defined in sample /etc/centrifydc/user.ignore and /etc/centrifydc/uid.ignore files. If you are not using group policies, you can uncomment the nss.user.ignore parameter in the /etc/centrifydc/centrifydc.conf file to ignore the default set of users.

> **Note**: If you plan to edit the user.ignore file, be sure to run the adreload command after modifying the file to have the changes take effect.

nss.user.ignore.all

This configuration parameter specifies how the list of users in nss.user.ignore is applied during lookups.

The parameter value can be true or false.

When you set this parameter to true, lookups generated by NSS, ldapproxy, or JAPI ignore the Active Directory users listed in nss.user.ignore.

When you set this parameter to false, only lookups generated by NSS ignore the Active Directory users listed in nss.user.ignore.

The default value is false.

nss.watch.slow.lookup.info.threshold

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request. When a NSS request exceeds this threshold, the NSS module writes information to a message to the INFO log file.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.info.threshold:10.

nss.watch.slow.lookup.info.threshold.group

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request categorized by group. When a NSS request exceeds this threshold for the user category, the NSS module writes to a message to the INFO log file.

The group category indicates the following NSS calls: getgrnam* getgrgid*

The nss.watch.slow.lookup.info.threshold.group setting overrides that set by the nss.watch.slow.lookup.info.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.info.threshold.group:35.

nss.watch.slow.lookup.info.threshold.user

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request categorized by user. When a NSS request exceeds this threshold for the user category, the NSS module writes to a message to the INFO log file.

The user category indicates the following NSS calls: getpwnam* getpwuid* getgrouplist

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the nss.watch.slow.lookup.info.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.info.threshold.user:15.

nss.watch.slow.lookup.warn.threshold

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.warn.threshold:20.

nss.watch.slow.lookup.warn.threshold.group

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request for the group category. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

The group category indicates the following NSS calls: getgrnam* getgrgid*

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the nss.watch.slow.lookup.warn.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.warn.threshold.group:30.

nss.watch.slow.lookup.warn.threshold.user

This configuration parameter specifies the threshold (in milliseconds) for the time spent on a complete NSS request for the user category. When a NSS request exceeds this threshold, the NSS module writes information to a WARN log file.

The user category indicates the following NSS calls: getpwnam* getpwuid* getgrouplist

The nss.watch.slow.lookup.info.threshold.user setting overrides that set by the nss.watch.slow.lookup.warn.threshold parameter.

By default, this parameter is set to -1, which means that there is no threshold.

For example: nss.watch.slow.lookup.warn.threshold.user:25.

lam.attributes.group.ignore

This parameter points to a file containing a list of AIX group attributes that the lam module should ignore and let AIX handle it to either provide the default value or return ENOATTR. The default is file:/etc/centrifydc/attributes.group.ignore.

lam.attributes.user.ignore

This parameter points to a file containing a list of AIX user attributes that the lam module should ignore and let AIX handle it to either provide the default value or return ENOATTR. The default is file:/etc/centrifydc/attributes.user.ignore.

lam.max.group.count

This configuration parameter applies to the AIX Loadable Authentication Module (LAM) and specifies the maximum number of Active Directory groups that the lsgroup ALL command will return.

The parameter value must be an integer. The default value for this parameter is 1000 groups. If you specify 0 or a negative value (for example, -1), there is no limit on the number of groups returned. For example:

lam.max.group.count: 100

Before changing this parameter setting or using a value of 0, you should consider its impact on your environment. Increasing the value of this parameter may provide more complete information about the number of Active Directory UNIX groups, but may result in slower performance if there are more Active Directory UNIX groups in the zone than the maximum you specify. Similarly, if you do not set a limit, you may experience performance problems if you have a large number of Active Directory groups. Decreasing the value of this parameter may provide better response time if there are more Active Directory UNIX groups in the zone than the maximum you specify, but further limits how much information is returned.

If this parameter is not defined in the configuration file, its default value is 1000 groups.

lam.max.user.count

This configuration parameter applies to the AIX Loadable Authentication Module (LAM) and specifies the maximum number of Active Directory users that the lsuser ALL command will return. This value also limit the results returned by the getpwent() and nextuser() functions.

The parameter value must be an integer. The default value for this parameter is 1000 users. If you specify 0 or a negative value (for example, -1), there is no limit on the number of users returned. For example:

lam.max.user.count: 100

Before changing this parameter setting or using a value of 0, you should consider its impact on your environment. Increasing the value of this parameter may provide more complete information about the number of Active Directory UNIX users, but may result in slower performance if there are more Active Directory UNIX users in the zone than the maximum you specify. Similarly, if you do not set a limit, you may experience performance problems if you have a large number of Active Directory users. Decreasing the value of this parameter may provide better response time if there are more Active Directory UNIX users in the zone than the maximum you specify, but further limits how much information is returned.

If this parameter is not defined in the configuration file, its default value is 1000 users.

Customizing NIS Configuration Parameters

This section describes the configuration parameters that affect the operation of the Delinea Network Information Service on the local host computer. The Delinea Network Information Service—`adnisd` -— provides a mechanism for responding to NIS client requests from computers not managed by a Server Suite Agent.

log.adnisd

This configuration parameter specifies the logging level for the Delinea Network Information Service. The default logging level is the logging level set for the log configuration parameter or INFO if neither parameter is defined in the configuration file. For example, to diagnose problems with the Delinea Network Information Service without changing the logging level for other components:

log.adnisd: DEBUG

log.adnisd.netgroup

This configuration parameter specifies the logging level for netgroup processing of the Centrify Network Information Service. The default logging level is the logging level set for the log.adnisd parameter if that parameter is defined. This parameter value can be set to DEBUG to log netgroup diagnostics or to INFO to suppress messages.

For example:

log.adnisd.netgroup: INFO

You can also set lower-level logging for netgroup processing using the following parameters:

log.adnisd.netgroup.syntax	Syntax warnings and errors for netgroup processing. The default value is the value defined for the log.adnisd.netgroup parameter.
log.adnisd.netgroup.inv	Inversion processing. The default value is the value defined for the log.adnisd.netgroup parameter. This parameter value can be set to DEBUG to log netgroup diagnostics or to INFO to suppress messages.

logger.facility.adnisd

This configuration parameter specifies the syslog facility to use for logging adnisd operations. This parameter enables you to log adnisd messages using a different syslog facility than the facilities used for logging general adclient messages or adclient audit messages. This parameter's value can be any valid syslog facility. For example, you can set this parameter to log messages to auth, authpriv, daemon, security, or localn facilities. The default is the auth facility. For example:

logger.facility.adnisd: auth

nisd.domain.name

This configuration parameter specifies the NIS domain name for the adnisd process to use when communicating with NIS clients.

For example, to specify that you want to use euro-all as the NIS domain name in the zone named Europe-00-Zone, you can set this parameter as follows:

nisd.domain.name: euro-all

If this parameter is not defined in the configuration file, the zone name is used by default.

nisd.exclude.maps

This configuration parameter specifies the name of the NIS maps you want to prevent the NIS service from using in response to NIS clients. This parameter enables you to exclude specific maps rather than explicitly specifying the maps you want to make available. For example, if you have a large number of automount maps or other network information that you want to make available to NIS clients but do not want to use agentless authentication, you can use this parameter to exclude the passwd and group maps but respond to automount or netgroup requests.

To use this configuration parameter, you must add the parameter name to the /etc/centrifydc.conf configuration file, then define its value. The parameter value must be a list of valid NIS map names, separated by spaces. For example:

nisd.exclude.maps: group passwd

This parameter excludes the named map and all derived maps. For example, if you specify group, the derived maps, group.byname, and group.bygid, are excluded. If this parameter is not defined in the configuration file, all NIS maps found in Active Directory are retrieved and available for service.

This configuration parameter overrides the setting of the nisd.maps parameter. If the same map is specified for both the nisd.exclude.maps and nisd.maps parameters, the map is excluded.

nisd.largegroup.name.length

This configuration parameter specifies the maximum number of characters to use in group names when groups with a large number of members are split into multiple new groups. Because some devices that submit NIS requests have limitations on the length of group names, you can use this parameter to specify the maximum length for group names.

When the adnisd process splits the group membership for a large group into multiple smaller groups, it truncates the original group name as needed to append the suffix defined in the nisd.largegroup.suffix parameter and not exceed the number of characters specified by this parameter. For example, if you have a large group named worldwide-all-corp, and have defined the suffix string as "-all" and the maximum length for group names as 10, when the worldwide-all-corp group membership is split into multiple groups, the groups are named as follows:

world-all1
world-all2
world-all3
world-all3

For example, to set the maximum length for group names to 20 characters:

nisd.largegroup.name.length: 20

If this parameter is not defined in the configuration file, the maximum group name length is 1024 characters by default.

nisd.largegroup.suffix

This configuration parameter specifies the suffix string or character to use in group names when automatically splitting up a group with large number of members.

Because group.bygid and group.byname NIS maps can often contain membership lists that exceed the 1024 limit for how much NIS data can be served to clients, the adnisd process will automatically truncate the membership list when this limit is reached. To allow the additional membership data to be retrieved, you can configure the Delinea Network Information Service to automatically split a large group into as many new groups as needed to deliver the complete membership list.

If you specify any value for the nisd.largegroup.suffix parameter, you enable the adnisd process to automatically split a large group into multiple new groups. When a group's data size exceeds 1024 data limit, a new group is created. The new group name is formed using the original group name, followed by the string defined for the nisd.largegroup.suffix parameter and ending in a number that represents the numeric order of the new group created.

For example, if you have a large group named performix-worldwide-corp, and have defined the suffix string as "-all" and the maximum length for group names as 10, when the performix-worldwide-corp group membership is split into multiple groups, the groups are named as follows:

performix-worldwide-corp-all1
performix-worldwide-corp-all2
performix-worldwide-corp-all3
performix-worldwide-corp-all4

All of the new groups have the same group identifier (GID) as the original group. If the new group names would exceed the maximum length for group names on a platform, you can use the nisd.largegroup.name.length parameter to set the maximum length for the new groups created.

If this configuration parameter is not set, the adnisd process truncates the group membership list such that each group entry is under 1024 characters.

nisd.maps

This configuration parameter specifies the name of the NIS maps currently available for NIS service. When the adnisd daemon connects to Active Directory, it retrieves the list of NIS maps available for the local computer's zone, creates a local map data store, and updates this configuration parameter, if necessary, to indicate the maps retrieved. If any NIS client requests a map that is not in the list specified by this parameter, the daemon refuses the request.

The parameter value must be a list of NIS map names. If the parameter is included in the configuration file but no value is set, no maps are retrieved from Active Directory or available for service.

For example, to make the netgroup maps available, but no other maps, you can set this parameter as follows:

nisd.maps: netgroup,netgroup.byhost,netgroup.byuser

> **Note**: You must specify all maps, including the derived maps.

If this parameter is not defined in the configuration file, all NIS maps found in Active Directory are retrieved and available for service.

nisd.maps.max

This configuration parameter specifies the number of alternate sets of NIS maps to retain. A new set of NIS maps is normally created when adnisd switches to an alternate domain controller. Keeping these alternate sets of maps allows Centrify Network Information Service to more efficiently switch between domain controllers.

The parameter value must be an integer greater than zero. The default is 2 map sets. For example:

nisd.maps.max: 2

nisd.net_addr

This configuration parameter sets the IP address the adnisd process uses for the NIS client socket. For example, the following sets the IP address to 192.168.212.11:

nisd.net_addr: 192.168.212.11

On systems with multiple Ethernet interfaces, adnisd configures RPC to the first interface. If an NIS client is trying to communicate on a different interface, adnisd will not receive the request.

Before creating sockets, adnisd reads centrifydc.conf file to see if an IP address and TCP and UPD ports are specified. If not, it uses localhost and random port numbers assigned by the operating system.

Use the nisd.port.udp and nisd.port.tcp parameters to complete the NIS port assignment.

nisd.passwd.expired.allow

This configuration parameter specifies whether a user with an expired Active Directory password should be allowed to log on to computers authenticated through NIS requests. The parameter value can be set to true or false.

By default, when a user's Active Directory password expires the password hash field in the passwd NIS map is replaced by two exclamation marks (!!), and the user is not allowed to log on to the local NIS client computer without first logging on to a Windows computer or an agent-managed computer running adclient to update the expired password. You can use this parameter to allow the user to log on locally using the expired password.

If you set the parameter value to true, users with an existing password hash in the passwd map generated from Active Directory do not have their password hash replaced by the exclamation marks and they can continue to log on using the expired password until they update their password in Active Directory. Once they update their password in Active Directory, in the NIS map is updated with a new password hash and users can log on with the new password. If a user never updates the Active Directory password by logging on to a Windows or agent-managed computer, however, the user's expired password may be used indefinitely.

The default value for this parameter is false. For example:

nisd.passwd.expired.allow: false

nisd.port.tcp

This configuration parameter sets the TCP port number the adnisd process uses to create the socket for NIS client communications. For example, the following sets the TCP port to 2556:

nisd.port.tcp: 2556

By default, no port number is specified. If you do not specify the port number, the operating system assigns a random port number.

Use the nisd.port.udp and nisd.net_addr parameters to complete the NIS client socket configuration.

nisd.port.udp

This configuration parameter sets the UDP port number the adnisd process uses to create the socket for NIS client communications. For example, the following sets the UPD port to 2555

nisd.port.udp: 2555

By default, no port number is specified. If you do not specify the port number, the operating system assigns a random port number.

Use the nisd.port.tcp and [nisd.net_addr]nisd-net-addr.md) parameters to complete the NIS client socket configuration.

nisd.securenets

This configuration parameter specifies a list of one or more subnets from which the daemon will accept NIS requests. You use this parameter to restrict access to the Delinea Network Information Service by IP address. NIS requests that do not come from the IP addresses specified in this configuration parameters are refused by the asnisd daemon.

> **Note**: You do not need to specify the local IP address for this parameter. The Delinea Network Information Service will always accept local NIS client requests.

The parameter value must include both the specific IP address or subnet and the subnet mask, separated by a forward slash. For example:

nisd.securenets: 192.168.111.0/255.255.255.0

You can specify multiple IP addresses by separating each IP address-subnet mask pair with a comma or a space. For example:

nisd.securenets: 192.68.11.0/255.255.255.0,192.147.10.0/255.255.255.0

If this parameter is not defined in the configuration file, only local NIS client requests are accepted by the asnisd process.

nisd.server.switch.delay

This configuration parameter specifies how long, in seconds, to wait before loading maps from a backup domain controller when the connection to the primary domain controller is lost. If the Delinea Network Information Service is unable to connect to its primary Active Directory domain controller, it will respond to NIS client requests using information in the local cache until the switch to the backup domain controller is complete.

The parameter value must be an integer equal to or greater than zero. If the value is zero, then the delay is disabled. For example, to set the delay period to 2 hours:

nisd.server.switch.delay: 7200

If this parameter is not defined in the configuration file, the default delay for switching to the backup domain controller is ten minutes (600 seconds).

nisd.startup.delay

This configuration parameter specifies the maximum number of seconds that the adnisd process should wait before responding to NIS client requests.

While adnisd retrieves and generates its NIS maps, it does not respond to client requests for the maximum number of seconds specified by this parameter. At the end of the startup delay time, adnisd will respond to NIS client requests whether all maps are loaded or not. Therefore, setting this parameter enables the adnisd process to begin responding to NIS clients requests before all NIS maps are loaded or created. You should be aware, however, that if the delay time is reached before all of the NIS maps are available, NIS clients may receive partial or empty answers to their requests.

> **Note**: If all of the NIS maps are loaded or created in less time than specified by this parameter, adnisd will begin responding to NIS requests without any startup delay.

By default, the maximum startup delay is 180 seconds. If you set this configuration parameter to zero, the adnisd process will only respond to NIS client requests after all NIS maps have been loaded or created. Therefore, in most cases, the parameter value should be a positive integer. For example, to set the startup delay to two minutes, you would set the parameter value to 120:

nisd.startup.delay: 120

nisd.threads

This configuration parameter specifies the maximum number of threads to allocate for processing NIS client requests.

The parameter value must be a positive integer within the valid range of 1 to 200. If you want to increase or decrease the number of threads used, you should stop the adnisd process, modify this parameter and save the configuration file, then restart the adnisd process.

The default value for this parameter is 4 threads. For example:

nisd.threads: 4

nisd.update.interval

This configuration parameter specifies the interval, in seconds, that the adnisd daemon waits between connections to Active Directory. At each interval, the adnisd daemon connects to Active Directory, gets the latest NIS maps for the local computer's zone, and updates its local NIS map data store.

The parameter value must be an integer equal to or greater than zero. If the value is zero, then the update interval is disabled and the local NIS map data store is not updated. For example, to set the interval for getting NIS maps to 1 hour:

nisd.update.interval: 3600

If this parameter is not defined in the configuration file, the default interval is 30 minutes (1800 seconds).

Customizing AIX Configuration Parameters

This section describes the configuration parameters that affect the administration of users and groups on AIX computers.

Setting extended attributes

AIX provides extended user and group attributes that enable administrators to specify user or group characteristics, such as the ability to login remotely to a user account, use the system resource controller (SRC) to execute programs, and so on. You can define these attributes for specific users and groups or for all user and group accounts on a local computer by editing specific configuration files such as /etc/security/user, /etc/security/group, and /etc/security/limits. The specific extended attributes available depend on the version of AIX you are using. For information about the extended attributes available for users and groups, see the AIX documentation for the security configuration files.

You can centralize administration of AIX computers by setting extended attributes for individual AIX users and groups in Active Directory. You can also set configuration parameters to set default extended attribute values for all Active Directory users or groups on a particular AIX computer.

> **Note**: Certain extended attributes, such as the system privileges, or capabilities attributes, are only supported by methods in the Loadable Authentication Module (LAM) version 5.2 or later.

The agent configuration file can include AIX configuration parameters that correspond to AIX extended attributes. For example:

admin	aix.user.attr.admin
daemon	aix.user.attr.daemon
rlogin	aix.user.attr.rlogin
su	aix.user.attr.su

Each configuration parameter has a hard-coded default value. You can edit the centrifydc.conf configuration file on any computer to change its default value. You should note that changes you make in the centrifydc.conf file only affect Active Directory users and groups. The settings do not affect local users or groups. Local users and groups get their extended attributes from the settings in the AIX configuration files, such as /etc/security/user and /etc/security/limits.

Enforcing access rights on AIX computers

If you are using the AIX Loadable Authentication Module (LAM), users who do not have the PAM login-all right can still log in. For example, an Active Directory user joined to the zone with the AIX computer and assigned to a role that does NOT include the login-all right can, in fact, log in to the AIX servers using the LAM interface. This is because the LAM interface does not use the rights defined in the user's Delinea role to control access. If the same server is configured with the PAM authentication module, that user would not be able to log in.

To control user log in activity, you have two choices:

- Keep the LAM interface and use one of the following PAM configuration parameters to define who has or does not have access:
  - pam.allow.groups: This configuration parameter specifies the groups allowed to access PAM-enabled applications.
  - pam.allow.users: This configuration parameter specifies the users who are allowed to access PAM-enabled applications.
  - pam.deny.groups: This configuration parameter specifies the groups that should be denied access to PAM-enabled applications.
  - pam.deny.users: This configuration parameter specifies the users that should be denied access to PAM-enabled applications.
- Replace the LAM interface with PAM. See the IBM AIX documentation for the instructions. The conversion procedure is fairly simple, however, you should test all applications on the server to ensure that they work the same with PAM. In addition, if you are using Delinea OpenSSH there are two versions: one for LAM and one for PAM. Both a LAM and PAM versions are distributed in the package. If you convert to PAM, uninstall the LAM version and install the PAM version.

Setting extended attributes

To set an extended attribute for an individual user, you can use adedit commands.

For example, to set the value of the extended attributes aix.ttys and aix.rlogin for the user joe, you might run commands similar to the following after binding to a domain and selecting a zone:

```
select_zone_user joe@ajax.acme.test
set_zone_user_field aix.ttys r1,r2,r3
set_zone_user_field aix.rlogin true
```

To verify the value of the extended attributes you have set, you might run commands similar to the following:

```
get_zone_user_field aix.ttys
r1,r2,r3
save_zone_user
```

You can also use adedit abbreviations to set and get extended attribute values. For example:

```
slzu joe@ajax.acme.test
szuf aix.fsize 209715
szuf aix.core 2097151
szuf aix.cpu -1
szuf aix.data 262144
```

Alternatively, you can also use configuration parameters to supplement the settings in the AIX /etc/security/user file. For example, if you have not explicitly defined the aix.rlogin attribute in /etc/security/user, you can set the following parameter in the centrifydc.conf file:

```
aix.user.attr.rlogin: false
```

You can use adquery and the keyword help to view a list of the supported extended attributes. For example:

```
adquery user --extattr help
```

aix.cache.extended.attr.enable

Use this parameter to specify whether to cache extended attribute default values. Caching extended attribute default values improves performance of the lsuser command.

The default value is false.

aix.user.attr.admgroups

This configuration parameter specifies the groups that the user account administers.

For the parameter value, enter a comma-separated list of groups; for example:

aix.user.attr.admgroups: unixAdmins,dnsAdmins

This parameter corresponds to the aix.admingroups attribute in the /etc/security.user file.

The default value is the empty string (no groups).

aix.user.attr.admin

This configuration parameter specifies the administrative status of the user.

Set the parameter value to true to define the user as an administrator; for example:

aix.user.attr.adm: true

Set the value to false to specify that the user is not an administrator. This is the default value.

This parameter corresponds to the aix.admin attribute in the /etc/security.user file.

aix.user.attr.auditclasses

This configuration parameter specifies the audit classes for the user.

You may enter a list of audit classes separated by commas, or the keyword ALL or an asterisk (*) to specify all audit classes. For example:

aix.user.attr.auditclasses: general,system

Place an exclamation point in front of a class to exclude it. For example, the following setting specifies all classes except system:

aix.user.attr.auditclasses: ALL,!system

This parameter corresponds to the aix.auditclasses attribute in the /etc/security.user file.

The default value is the empty string (no audit classes).

aix.user.attr.core

This configuration parameter specifies the soft limit for the largest core file that the user can create. Use -1 to set an unlimited size.

For example, to set the value to 2097151:

aix.user.attr.core: 2097151

This parameter corresponds to the aix.core attribute in the /etc/security.limits file.

The default value is 2097151.

aix.user.attr.cpu

This configuration parameter specifies the soft limit (in seconds) for the amount of system time that a user's process can use.

Use -1 to set an unlimited size; for example, to set the limit to one hour:

aix.user.attr.cpu: 3600

This parameter corresponds to the aix.cpu attribute in the /etc/security.limits file.

The default value is -1.

aix.user.attr.data

This configuration parameter specifies the soft limit for the largest data-segment for a user's process. Use -1 to set an unlimited size.

For example, to se the value to 2097151:

aix.user.attr.data: 2097151

This parameter corresponds to the aix.data attribute in the /etc/security.limits file.

The default value is 2097151.

aix.user.attr.daemon

This configuration parameter specifies whether the user can execute programs using the system resource controller (SRC), which manages system daemons and sub systems such as adclient and NFS.

Set the parameter value to true to allow users to execute programs using the SRC. Set the value to false to prevent users from executing programs using the SRC.

This parameter corresponds to the aix.daemon attribute in the /etc/security.user file.

The default value is false, which prevents a user from executing programs using SRC.

aix.user.attr.fsize

This configuration parameter specifies the soft limit for the largest file that the user process can create. Use -1 to set an unlimited size.

For example, to set the value to 2097151:

aix.user.attr.fsize: 2097151

This parameter corresponds to the aix.fsize attribute in the /etc/security.limits file.

The default value is 2097151.

aix.user.attr.nofiles

This configuration parameter specifies the soft limit for the number of file descriptors that the user's process may have open at one time.

Use -1 to set an unlimited size.

For example, to set the limit to 2000:

aix.user.attr.nofiles: 2000

This parameter corresponds to the aix.nofiles attribute in the /etc/security.limits file.

The default value is -1.

aix.user.attr.nprocs

This configuration parameter specifies the soft limit on the number of processes a user can have running at one time.

Use -1 to specify the maximum number allowed by the system; for example:

aix.user.attr.nprocs: -1

This parameter corresponds to the aix.nprocs attribute in the /etc/security/limits file.

The default value is -1.

aix.user.attr.rlogin

This configuration parameter specifies whether remote users can access the user account through rlogin and telnet.

Set the parameter value to true to allow remote access to the user account.

Set the parameter value to false to prevent remote access to the user account.

This parameter corresponds to the aix.rlogin attribute in the /etc/security.user file.

The default value is true, which allows remote access to the user account.

aix.user.attr.rss

This configuration parameter specifies the soft limit for the largest amount of system memory that the user process can allocate. Use -1 to set an unlimited size.

For example, to set the value to 2097151:

aix.user.attr.rss: 2097151

This parameter corresponds to the aix.rss attribute in the /etc/security.limits file.

The default value is 65536.

aix.user.attr.stack

This configuration parameter specifies the soft limit for the largest stack segment for the user's process. Use -1 to set an unlimited size.

For example, to set the value to 2097151:

aix.user.attr.stack: 2097151

This parameter corresponds to the aix.stack attribute in the /etc/security.limits file.

The default value is 65536.

aix.user.attr.su

This configuration parameter specifies whether other users can use the su command to switch to this user account.

Set the parameter value to true to allow other users to switch to this user account.

Set the value to false to prevent users from switching to this user account.

This parameter corresponds to the aix.su attribute in the /etc/security.user file.

The default value is true, which allows other users to switch to this user account.

aix.user.attr.sugroups

This configuration parameter specifies the groups that can use the su command to switch to this user account.

You may enter a list of groups separated by commas, or the keyword ALL or an asterisk (*) to specify all groups. For example:

aix.user.attr.sugroups: admins,unixAdmins,dnsAdmins,enterpriseAdmins

Place an exclamation point in front of a group to exclude it. For example, the following setting specifies all groups except dnsAdmins:

aix.user.attr.sugroups: ALL,!dnsAdmins

This parameter corresponds to the aix.sugroups attribute in the /etc/security.user file.

The default value is ALL, which allows all groups to switch to the user account.

aix.user.attr.threads

This configuration parameter specifies the soft limit for the largest number of threads that a user process can create.

Use -1 to specify an unlimited number; for example:

aix.user.attr.threads: -1

This parameter corresponds to the aix.thread attribute in the /etc/security/limits file.

The default value is -1, which specifies an unlimited number of threads.

aix.user.attr.tpath

This configuration parameter specifies the status of the user's trusted path. The trusted path prevents unauthorized programs from reading data from the user terminal.

Set one of the following values for this parameter:

always	Allows the user to execute trusted processes only, which means the that the user's initial program must be in the trusted shell or another trusted process.
notsh	Prevents the user from invoking the trusted shell on a trusted path. Entering the secure attention key (SAK) causes the login session to terminate.
nosak	Disables the secure attention key (SAK) for all processes run by the user. Specify nosak if the user transfers binary that may contain the SAK. This is the default value.
on	Provides the user with normal trusted path characteristics; the user can invoke a trusted path (enter a trusted shell) with the secure attention key (SAK).

This parameter corresponds to the aix.tpath attribute in the /etc/security.user file.

The default value is nosak.

aix.user.attr.ttys

This configuration parameter specifies the terminals that can access the user account.

You may enter a list of terminals separated by commas, or the keyword ALL or an asterisk (*) to specify all terminals. For example:

aix.user.attr.ttys: /dev/pts

> **Note**: You must specify /dev/pts or ALL for network logins to work.

Place an exclamation point in front of a group to exclude it.

This parameter corresponds to the aix.ttys attribute in the /etc/security.user file. The default value is ALL, which allows all terminals to access the user account.

aix.user.attr.umask

This configuration parameter specifies the default umask to define permissions for the user. The umask value along with the permissions of the creating process determine the permissions for a new file.

This parameter corresponds to the aix.ttys attribute in the /etc/security.user file.

The parameter value can be set to a three-digit octal value. The default value is 022.

Customizing Delinea UNIX programs Configuration Parameters

This section describes the configuration parameters that affect the operation of Delinea UNIX command line programs on the local host computer.

adjoin.adclient.wait.seconds

This configuration parameter specifies the number of seconds the adjoin command should wait before exiting to ensure that the agent is available to complete the join operation.

For example, to configure the adjoin command to wait 10 seconds:

adjoin.adclient.wait.seconds: 10

adjoin.krb5.conf.file

This configuration parameter specifies the path to a customized Kerberos configuration file you want to use to join a domain.

The parameter value must be a path name. For example:

adjoin.krb5.conf.file: /etc/centrifydc/krb5_join.conf

adjoin.samaccountname.length

This configuration parameter specifies the maximum number of characters to use when the adjoin command must generate a pre-Windows 2000 computer name by truncating the host name. This parameter also determines how adjoin creates the computer account in Active Directory.

The default value is 15 characters to conform to the maximum length allowed by the NetLogon service, which is the preferred service for adclient to use for NTLM pass-through authentication. NetLogon is fast and automatically returns a user's group membership.

The maximum length allowed for the pre-Windows 2000 computer name, which is stored in the sAMAccountName attribute for the computer account in Active Directory, is 19 characters. However, if you specify more than 15 characters (up to the 19 character limit) adclient will use slower NTLM authentication methods, and will use additional LDAP searches to fetch the user's group membership.

> **Note**: This configuration parameter is ignored if you run the adjoin command with the --prewin2k option to manually specify the pre-Windows 2000 computer name.

The parameter value should be a positive integer in the valid range of 1 to 19 characters. For example:

adjoin.samaccountname.length: 15

If you specify a value greater than 19, the parameter setting is ignored and the computer name is truncated at 19 characters in the sAMAccountName attribute for the computer account.

If the computer's host name size exceeds the specified value for this parameter, adjoin will use LDAP (and require administrative privileges) to create computer accounts, instead of MS-RPC. In any case, if the computer's short host name exceeds 19 characters, then it is no longer possible to create computer accounts by using MS-RPC methods and LDAP will be used instead.

adpasswd.account.disabled.mesg

This configuration parameter specifies the message displayed by the adpasswd program when users cannot change their password because their account is locked.

For example:

adpasswd.account.disabled.mesg:
Account cannot be accessed at this time. Please contact your system administrator.

adpasswd.account.invalid.mesg

This configuration parameter specifies the message displayed by the adpasswd program when a user account is unrecognized or the password is invalid.

For example:

```
adpasswd.account.invalid.mesg: \
Invalid username or password
```

adpasswd.password.change.disabled.mesg

This configuration parameter specifies the message displayed by the adpasswd program when users are not allowed to change their password because password change for these users has been disabled in Active Directory.

For example:

adpasswd.password.change.disabled.mesg: \
Password change for this user has been disabled in Active Directory

adpasswd.password.change.perm.mesg

This configuration parameter specifies the message displayed by the adpasswd program when a user cannot change another user's password because of insufficient permissions.

For example:

adpasswd.password.change.perm.mesg:
You do not have permission to change this users password. Please contact your system administrator.

Customizing Smart Card Configuration Parameters

This section describes the configuration parameters that affect the use of Delinea access control smart cards on the local host computer.

smartcard.allow.noeku

This configuration parameter allows the use of certificates that do not have the Extended Key Usage (EKU) attribute. Normally, smart card use requires certificates with the EKU attribute. The value of this parameter can be true or false.

If you set this parameter to true, certificates without an EKU attribute can be used for SmartCard logon, and certificates with the following attributes can also be used to log on with a smart card:

- Certificates with no EKU
- Certificates with an All Purpose EKU
- Certificates with a Client Authentication EKU

If you set this parameter to false, only certificates that contain the smart card logon object identifier can be used to log on with a smart card. The default value of this parameter is false.

After changing the value of this parameter, you must re-enable smart card support by running the following sctool command as root:

[root]$ sctool -E

When you run sctool with the -E option, you must also specify the -a or -k option. You can also control this feature using group policy.

smartcard.login.force

This configuration parameter forces users to log in with a smart card.

By default, this parameter is set to `False`, which means that users are not forced to use a smart card (it's optional).

To force the use of smart cards, set this parameter to `True`:

```
smartcard.login.force: true
```

smartcard.name.mapping

This configuration parameter turns on support for multi-user smart cards.

By default, this parameter is set to False, which prevents the use of multi-user smart cards.

To allow the use of multi-user smart cards, set this parameter to True:

smartcard.name.mapping: true

smartcard.pkcs11.module

This configuration parameter specifies the path to the PKCS #11 module to be used by smart card components on the computer.

By default, smart card components use the Delinea Coolkey PKCS #11 module. However, Coolkey does not support all smart cards so you may specify a different module if necessary by specifying the absolute path to your PKCS #11 module with this parameter. For example:

rhel.smartcard.pkcs11.module /usr/$LIB/pkcs11/opensc-pkcs11.so

> **Note**: In the path specification, this parameter supports the use of the $LIB environment variable, which allows a single path specification to work for 32-bit and 64-bit systems. At run time, on 32-bit systems, $LIB resolves to lib, while on 64-bit systems it resolves to lib64.

After changing the value of this parameter, you must re-enable smart card support by running the following sctool commands as root:

```
[root]$ sctool --disable
[root]$ sctool --enable
```

Also, refresh the GNOME desktop by running the following command as root:

```
[root]$ systemctl restart gdm
```

In most cases, you set this configuration parameter using group policy.

rhel.smartcard.pkcs11.module (Deprecated)

This configuration parameter specifies the path to the PKCS #11 module to be used by smart card components on the computer.

By default, smart card components use the Delinea Coolkey PKCS #11 module. However, Coolkey does not support all smart cards so you may specify a different module if necessary by specifying the absolute path to your PKCS #11 module with this parameter. For example:

rhel.smartcard.pkcs11.module /usr/$LIB/pkcs11/opensc-pkcs11.so

> **Note**: In the path specification, this parameter supports the use of the $LIB environment variable, which allows a single path specification to work for 32-bit and 64-bit systems. At run time, on 32-bit systems, $LIB resolves to lib, while on 64-bit systems it resolves to lib64.

After changing the value of this parameter, you must re-enable smart card support by running the following sctool commands as root:

```
[root]$ sctool --disable
[root]$ sctool --enable
```

Also, refresh the GNOME desktop by running the following command as root:

```
[root]$ /usr/sbin/gdm-safe-restart
```

In most cases, you set this configuration parameter using group policy.

Customizing Authorization Configuration Parameters

This section describes the configuration parameters that affect the operation of authorization features (privilege elevation service) on the local host computer. You can configure authorization rules by defining specific command- or application-level rights, combining those rights into roles, and assigning users to those roles to control the operations they are allowed to perform on specific computers in a zone.

Delinea

adclient.azman.refresh.interval

This configuration parameter is deprecated and is replaced by the adclient.refresh.interval.dz parameter. See adclient.refresh.interval.dz for details about adclient.refresh.interval.dz.

The Server Suite upgrade utility renames this parameter if it is being used.

adclient.cache.flush.interval.dz

This configuration parameter specifies the frequency (in seconds) with which the Delinea Agent for *NIX flushes its authorization cache. You should note that this parameter only forces periodic updates to the authorization cache. It does not affect the agent's primary domain controller cache.

The default value is 0, which completely disables periodic flushing of the authorization cache.

The parameter value must be a positive integer. For example, to force the authorization cache to be cleared every 30 minutes, set the parameter as follows:

adclient.cache.flush.interval.dz: 1800

adclient.dz.refresh.hook

This parameter specifies the full path of the command that will be executed after adclient finishes the authorization cache refresh. By default, the value of this parameter is as follows:

adclient.dz.refresh.hook: /usr/share/centrifydc/etc/hooks/dz_refresh

The dz_refresh script will check for and remove any expired files in the /var/centrifydc/dz.ovr.d directory. Files in that directory have filenames in the format of yymmddHHMMSS-*.ovr, where the yymmddHHMMSS is the expiration date and time.

You can configure how often the authorization cache refreshes with the adclient.refresh.interval.dz parameter.

adclient.dzdo.clear.passwd.timestamp

This configuration parameter specifies whether users must re-authenticate with dzdo after logging out.

When a user authenticates with dzdo, a ticket is temporarily created that allows dzdo to run without re-authentication for a short period of time (set by the dzdo.timestamp_timeout parameter). If a user logs out, the ticket is reused when the user logs back in.

The parameter value can be true or false. Setting this parameter to true clears the ticket and requires users to re-authenticate to use dzdo after logging out and back in. The default parameter value is false.

For example:

adclient.dzdo.clear.passwd.timestamp: true

You can also set this parameter using group policy.

adclient.refresh.interval.dz

Note: Starting with agent version 5.1.3, this configuration parameter replaces the deprecated adclient.azman.refresh.interval parameter.

This configuration parameter specifies the maximum number of minutes to keep access control information from the authorization store cached before refreshing the data from Active Directory. Access control information consists of rights, roles, and role assignments that the Delinea Privilege Elevation Serviceuses to control access to dzdo privileged commands, dzsh restricted environments, PAM-enabled applications, and some third-party application.

Because the agent handles connecting to and retrieving information from Active Directory, this configuration parameter controls how frequently adclient checks for updates to the privilege elevation service set of information from Active Directory. If any privilege elevation service information has been modified, the cache is refreshed with the new information.

If local account management is enabled, this configuration parameter also specifies how often etc/group and etc/passwd are updated on individual computers based on the local group and local user settings that you configure in Access Manager.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

If you are manually setting this parameter, the parameter value must be a positive integer. The following example sets the cache expiration time to 30 minutes:

adclient.refresh.interval.dz: 30

If this parameter is not defined in the configuration file, its default value is 30 minutes.

adclient.sudo.clear.passwd.timestamp

This configuration parameter is used together with the tty_tickets parameter in the sudoers configuration file (/etc/sudoers) to specify whether users must re-authenticate with sudo after logging out.

When a user authenticates with sudo, a ticket is temporarily created that allows sudo to run without re-authentication for a short period of time. If a user logs out and the ticket is not cleared, the ticket is reused when the user logs back in, and the user does not need to re-authenticate. If a user logs out and the ticket is cleared, the user must re-authenticate with sudo when logging back in.

Starting with release 2015, the way that you configure whether re-authentication is required depends on the tty_tickets parameter in the sudoers configuration file (/etc/sudoers.conf). In some situations, re-authentication requirements are also controlled by this parameter. Details are as follows:

- If tty_tickets is enabled, tickets are always removed when a sudo user logs out, regardless of whether this parameter is set to true or false. That is, when tty_tickets is enabled, this parameter has no effect, and sudo users must always re-authenticate.
- If tty_tickets is disabled, the requirement for sudo users to re-authenticate is controlled by this parameter and the **Force sudo re-authentication when relogin** group policy.

Tickets are cleared, and sudo re-authentication is required, under these scenarios:

- The tty_ticket parameter in the sudoers configuration file is enabled (it is enabled by default), or
- The tty_ticket parameter in the sudoers configuration file is disabled and the adclient.sudo.clear.passwd.timestamp parameter is set to true, or
- The tty_ticket parameter in the sudoers configuration file is disabled and the **Force sudo re-authentication when relogin** group policy is enabled.

Tickets are not cleared, and sudo re-authentication is not required, under these scenarios:

- The tty_ticket parameter in the sudoers configuration file is disabled and the adclient.sudo.clear.passwd.timestamp parameter is set to false, or
- The tty_ticket parameter in the sudoers configuration file is disabled and the **Force sudo re-authentication when relogin** group policy is disabled.

The default parameter value is false.

For example:

adclient.sudo.clear.passwd.timestamp: false

You can also set this parameter using group policy.

adclient.sudo.timestampdir

This configuration parameter specifies the directory where authentication tickets reside. By default, the directory is /var/run/sudo. Some platforms use a different directory for tickets, such as /var/db/sudo/user (RHEL) or /var/lib/sudo/user (Ubuntu), which you can specify in this parameter.

The default value of this parameter is /var/run/sudo.

For example:

adclient.sudo.timestampdir: /var/run/sudo

audittrail.dz.command.with.args

This configuration parameter specifies whether to show command parameters in the audit log for dzdo and dzsh or just the command name. The default (false) is to show only the command name. For example, to keep passwords entered on the command line out of the log, leave this parameter set to false.

Set to true to show the command parameters as well as the command name.

For example:

audittrail.dz.command.with.args: true

dz.auto.anchors

This configuration parameter specifies whether you want to add anchors ($) automatically to the regular expressions you define as command rights and use in role definitions.The default setting is true to avoid matching unintended paths or commands if the regular expression pattern is not carefully set. If you set this parameter to false, you should carefully review all regular expressions used as command rights to identify all possible matches for the pattern defined.

For example:

dz.auto.anchors: true

dz.enabled

This configuration parameter is only applicable for classic zones to specify whether authorization services are enabled or disabled. In hierarchical zones, which must have agents version 5.x or later, this parameter is not applicable and is ignored. In classic zones, however, authorization is an optional feature that can be explicitly enabled or disabled.

In classic zones, users can log on as long as they have a profile in a zone. In hierarchical zones, users must be assigned to a role that grants them permission to log on. If you have agents that are joined to a classic zone, you can set this parameter to false to explicitly prevent the agent from looking up authorization information to reduce network traffic.

If you have agents from version 4.x, the default value for this parameter is true. This parameter is not defined for agents version 5.x and later.

For example:

dz.enabled: false

dz.system.path

This configuration parameter specifies the list of common System paths for locating commands in the local operating environment. The paths specified for this parameter define the program locations searched when the System match path option is selected for dzdo and dzsh commands.

This configuration parameter enables an administrator to define rights to run commands found in the user's path, the system path, or a specific location, even though the default or most commonly used paths may be different in different operating environments.

The default value for this parameter lists the most common locations for finding command line programs in the system path. For example:

dz.system.path: "/sbin:/usr/sbin:/usr/local/sbin"

dz.user.path

This configuration parameter specifies the list of common User paths for locating commands in the local operating environment. The paths specified for this parameter define the program locations searched when the User match path option is selected for dzdo and dzsh commands.

This configuration parameter enables an administrator to define rights to run commands found in the user's path, the system path, or a specific location, even though the default or most commonly used paths may be different in different operating environments.

The default value for this parameter lists the most common locations for finding command line programs in the user's path. For example:

dz.user.path: "/bin:/usr/bin:/usr/local/bin"

dzdo.always_set_home

This configuration parameter specifies whether privileged commands run with dzdo commands should set the HOME environment variable to the home directory of the target user (which is root by default). The parameter value can be true or false. Setting this parameter to true effectively implies that the -H command line option should always used. The default parameter value is false.

For example:

dzdo.always_set_home: false

This configuration parameter provides functionality equivalent to the always_set_home flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.badpass_message

This configuration parameter specifies the message that should be displayed if a user enters an incorrect password. The parameter value can be any text string enclosed by quotation marks.

For example:

dzdo.badpass_message: “The password provided is not valid.”

The default value is “Sorry, try again.”

This configuration parameter provides functionality equivalent to the badpass_message flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.command_alias

This configuration parameter specifies a mapping file containing mappings between command aliases and command files for all of the command aliases that a customer uses. If you specify a mapping file in dzdo.command_alias and then issue a dzdo command using a command alias, dzdo searches the mapping file to see if the first dzdo parameter matches any of the aliases.

If there is a match, the command path specified for the alias in the mapping file is used by dzdo to perform command matching to determine whether the command is allowed to run.

The parameter value has the following syntax:

dzdo.command_alias: *aliasfile_full_pathname*

For example, the following line in centrifydc.conf results in the default mapping file (dzdo.commandalias.map) being used:

dzdo.command_alias: /etc/centrifydc/dzdo.commandalias.map

The syntax of the content within the mapping file is:

command_alias_1: command_path [*arguments*]
command_alias_n: command_path [*arguments*]

For example, a mapping file could contain the following, which defines two command aliases—oracle_startup and centrifydc_startup:

oracle_startup: /opt/oracle/startup
centrifydc_startup: /opt/centrifydc/startup

Actual mapping files can contain any number of aliases.

dzdo.edit.checkdir

This configuration parameter prevents a user from editing files using the dzedit command in a directory that the user already has permissions to edit using their own account.

If a user who can already write to a directory with their own account uses dzedit to edit the same directory, it is possible that they may unintentionally edit arbitrary files in the directory if wild cards are used to specify the files the user intends to edit.

Set this configuration parameter to true to specify whether dzedit will check the user's directory permissions, and deny the user the ability to modify files in the directory when they run as root if they have sufficient permissions to edit the directory using their own account.

For example:

dzdo.edit.checkdir: true

The default value of this configuration parameter is true.

dzdo.edit.follow

This configuration parameter prevents users from editing a file in a directory using dzedit that is reached by following a symbolic link (symlink) if the user already has permissions to edit the directory. Edits are also prevented on all sub-directories on the file path.

In some cases, if a user that already has permissions to write to a directory but invokes dzedit to edit a file in that directory which contains a symlink, they may edit the linked file as well.

If set to false, this configuration parameter will not allow a user to edit files reached by symbolic link by using dzedit.

It is strongly recommended that you keep the specified value as false.

For example:

dzdo.edit.follow: false

The default value of this configuration parameter is false.

dzdo.env_check

This configuration parameter specifies the list of environment variables that the dzdo process should check for the special characters, % or /, in the value. If the dzdo process finds environment variable values containing the special characters, it removes those variables from the user's environment. Variables with % or / characters are removed regardless of whether you have selected the **Reset environment variables** option for the command in Access Manager.

The default list of variables to check is displayed when you run dzdo -V command as root. You can customize the list by modifying this configuration parameter in the centrifydc.conf file.

The parameter value can be a comma-separated list of environment variable names.

For example:

dzdo.env_check: COLORTERM,LANG,LANGUAGE,LC_*,LINGUAS,TERM

This configuration parameter provides functionality equivalent to the env_reset flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.env_delete

This configuration parameter specifies the default list of environment variables to be removed from the user's environment. This configuration parameter only applies if you have selected the **Remove unsafe environment variables** option for the command in the Access Manager. The variables specified with this parameter are removed in addition to the default list of variables displayed when you run the dzdo -V command as root.

The parameter value can be a comma-separated list of environment variable names.

For example:

dzdo.env_delete: IFS,CDPATH,LOCALDOMAIN,RES_OPTIONS,HOSTALIASES, \ NLSPATH,PATH_LOCALE,LD_*,*RLD**,*TERMINFO,TERMINFO_DIRS,* \ *TERMPATH,TERMCAP,ENV,BASH_ENV,PS4,GLOBIGNORE,SHELLOPTS,* \ *JAVA_TOOL_OPTIONS,PERLIO_DEBUG,PERLLIB,PERL5LIB,* \ *PERL5OPT,PERL5DB,FPATH,NULLCMD,READNULLCMD,ZDOTDIR,TMPPREFIX,* \ *PYTHONHOME,PYTHONPATH,PYTHONINSPECT,RUBYLIB,RUBYOPT,KRB5_CONFIG,* \ *KRB5_KTNAME,VAR_ACE,USR_ACE,DLC_ACE,SHLIB_PATH,LDR**, \ LIBPATH,DYLD_*

This configuration parameter provides functionality equivalent to the env_delete flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.env_keep

This configuration parameter specifies the default list of environment variables to preserve in the user's environment. This configuration parameter only applies if you have selected the **Reset environment variables** option for the command in the Access Manager. The variables specified with this parameter are preserved in addition to the default list of variables displayed when you run the dzdo -V command as root.

The parameter value can be a comma-separated list of environment variable names.

For example:

dzdo.env_keep: COLORS,DISPLAY,HOME,HOSTNAME,KRB5CCNAME, LS_COLORS,MAIL,PATH,PS1,PS2,TZ,XAUTHORITY,XAUTHORIZATION

This configuration parameter provides functionality equivalent to the env_keep flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.lecture

This configuration parameter specifies whether dzdo displays a warning message about using the program before displaying the password prompt. The valid parameter values are:

once	To display the warning message only the first time the command is run.
never	To never display a warning message.
always	To display the warning message every time the program is invoked.

The default parameter value is once. For example:

dzdo.lecture: once

This configuration parameter provides functionality equivalent to the lecture flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.lecture_file

This configuration parameter specifies the full path to a file containing the warning message you want displayed. If this parameter is not set, a default message is displayed.

For example, to use a custom message in the file dzdo_warning:

dzdo.lecture_file: /etc/custom/dzdo_warning

This configuration parameter provides functionality equivalent to the lecture_file flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.legacyzone.mfa.enabled

Enable this configuration parameter to require multi-factor authentication for users to run the dzdo command. If you enable this parameter, users will be required to authenticate with MFA if they are required to re-authenticate to run dzdo, and are listed in either adclient.legacyzone.mfa.required.users or adclient.legacyzone.mfa.required.groups.

You must enable adclient.legacyzone.mfa.enabled for this policy to take effect.

This configuration parameter does not support rescue rights; users listed in adclient.legacyzone.mfa.rescue.users will not be able to run dzdo without MFA.

To enable this policy, set this parameter to true. The default value for this parameter is false.

For example:

dzdo.legacyzone.mfa.enabled: true

dzdo.log_good

This configuration parameter specifies whether you want to log messages for successful command execution. By default, the dzdo program logs both valid and invalid command execution. To log information about only invalid command execution, set this parameter to false. The default value for this parameter is true.

For example:

dzdo.log_good: true

The dzdo program typically logs messages to the file /var/log/secure.

You can also set this parameter using group policy.

dzdo.passprompt

This configuration parameter lets you specify the password prompt displayed when running privileged commands. This parameter serves the same function as the dzdo -p command.

You can use the following escapes in the prompt:

%u	Expands to the invoking user's login name
%U	Expands to the login name of the user the command will be run as. If not specified, defaults to root
%h	Expands to the local hostname without the domain name
%H	Expands to the local hostname including the domain name
%p	Expands to the user whose password is asked for
%%	Collapses to a single % character

The default prompt is [dzdo] password for %p: where %p is root unless specified otherwise.

For example,

dzdo.passprompt: "[dzdo] Enter password for %U@%h"

You can also set this parameter using group policy.

dzdo.passwd_timeout

This configuration parameter specifies the number of minutes before the dzdo password prompt times out. The default parameter value is 5 minutes. You can set this parameter to zero (0) to have the password prompt never timeout.

For example:

dzdo.passwd_timeout: 5

This configuration parameter provides functionality equivalent to the passwd_timeout flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.path_info

This configuration parameter specifies whether the dzdo program should inform the user when it cannot find a command in the user's PATH. By default, the parameter value is true and the program will display an error statement indicating that the command could not be found in the user's PATH. You can set this configuration parameter to false if you want to prevent dzdo from indicating whether a command was not allowed or simply not found.

For example:

dzdo.path_info: true

This configuration parameter provides functionality equivalent to the path_info flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.search_path

This configuration parameter specifies the search path for the dzdo program to use to look for commands and scripts that require privileges to run. You can specify a list of directories for the dzdo program to search for commands and scripts. If you configure this parameter, the dzdo program will search in the specified directories no matter which path the command rights are configured to use in the Access Manager **System search path** option.

If commands are configured to use the path defined in the Access Manager **System search path** option and the dzdo.search_path parameter is not defined, the following actions take place:

- The current user's path is used to search for the commands.
- Only the commands located under the System path are allowed to execute.

There is no default value for this parameter.

The parameter value can be a list of directories or the name of a file that contains the list of directories. For example, you can specify a file that contains the directories to search using the file: keyword and a file location:

dzdo.search_path: file:/etc/centrifydc/customized_dzdo_directories

If you specify a file name for this parameter, you should ensure the file is owned by root and not accessible to any other users.

You can also set this parameter using group policy.

dzdo.requiretty

This configuration parameter specifies whether a user needs to be logged in to a valid tty session in order to run dzdo. If you set this parameter to true, this means that the user can run dzdo only from a login session and not from a cgi-bin or cron(8) script.

By default, this parameter is set to false so that it's not required to run a tty session in order to run dzdo.

dzdo.secure_path

This configuration parameter specifies the path for the dzdo program to use when executing commands and scripts that require privileges to run. If you specify a directory using this parameter, the dzdo program will only execute commands and scripts that are found in that directory.

Setting both the dzdo.search_path and dzdo.secure_path parameters to the same value is equivalent to setting the secure_path parameter in the sudoers configuration file.

There is no default value for this parameter.

The parameter value can be a list of directories or the name of a file that contains the list of directories. For example, you can specify a file that contains the directories to search using the file: keyword and a file location:

dzdo.secure_path: file:/etc/centrifydc/customized_dzdo_directories

Within the file, lines should contain path separated by colons. For example, a file specifying two paths might look like this:

/etc/centrifydc/reports/exec_report_cmds:/usr/sbin/ora_cmds

If you specify a file name for this parameter, you should ensure the file is owned by root and not accessible to any other users.

You can also set this parameter using group policy.

dzdo.set_home

This configuration parameter sets the HOME environment variable to the home directory of the target user when the -s command line option is used. The parameter value can be true or false. The default parameter value is false.

For example:

dzdo.set_home: false

This configuration parameter provides functionality equivalent to the set_home flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.set.runas.explicit

This configuration parameter specifies whether a user must explicitly identify the 'runas' user when executing a command with dzdo.

The parameter value can be true or false; the default value is true.

When the parameter value is true, if a user executes a command with dzdo and does not explicitly identify the user or group to run as (with the -u or -g option), adclient assumes that the command should be run as root. If the user is not authorized to run the command as root, dzdo fails to execute the command and issues an error message; for example:

```
User u1 is authorized to run adinfo as user qa1
dzdo.set.runas.explicit: true
...
[u1@rh6]$dzdo adinfo
Sorry, user u1 is not allowed to execute '/usr/bin/adinfo' as root on rh6.
```

When the parameter value is false, if a user executes a command with dzdo and does not explicitly identify the user or group to run as (with the -u or -g option), adclient attempts to resolve the user. If the command defines a single runas user, dzdo executes the specified command and sends a message to the log file; for example:

```
User u1 is authorized to run adinfo as user qa1
dzdo.set.runas.explicit: false
...
[u1@rh6]$dzdo adinfo
Local host name: rh6
Joined to domain acme.com
...
```

If the command defines multiple runas users, dzdo cannot resolve the user to run as and attempts to run the command as root. Since the user is not authorized to run the command as root, dzdo fails to execute the command and issues an error message; for example:

```
User u1 is authorized to run adinfo as users qa1 and adm
dzdo.set.runas.explicit: true
...
[u1@rh6]$dzdo adinfo
Sorry, user u1 is not allowed to execute '/usr/bin/adinfo' as root on rh6.
```

In all cases, a user can execute a command successfully with dzdo by using the -u option to explicitly identify the runas user; for example:

```
[u1@rh6]$dzdo -u qa1 adinfo
Local host name: rh6
Joined to domain acme.com
...
```

You can also set this parameter using group policy.

dzdo.timestampdir

This configuration parameter specifies the directory where dzdo stores the user's login timestamp files. The default is directory is /var/run/dzdo.

For example:

dzdo.timestampdir: /var/run/dzdo

This configuration parameter provides functionality equivalent to the timestampdir flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.timestamp_timeout

This configuration parameter specifies the maximum number of minutes allowed between operations before prompting the user to re-enter a password. The default parameter value is 5 minutes. You can set this parameter to zero (0) to always prompt for a password when users run privileged commands with dzdo. If set to a value less than 0 the user's timestamp never expires.

For example:

dzdo.timestamp_timeout: 5

This configuration parameter provides functionality equivalent to the timestamp_timeout flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.timestamp_type

The privilege elevation service uses per-user timestamp files for credential caching. You can use this configuration parameter to specify the type of timestamp record for the service to use.

By default, this parameter is set to tty.

You can set this parameter to any of the following values:

- **global**: A single time stamp record is used for all of a user's login sessions, regardless of the terminal or parent process ID.
- **ppid**: A single time stamp record is used for all processes with the same parent process ID (usually the shell). Commands run from the same shell (or other common parent process) will not require a password for dzdo.timestamp_timeout minutes (5 by default). Commands run by way of sudo with a different parent process ID, for example from a shell script, will be authenticated separately.
- **tty**: One time stamp record is used for each terminal, which means that a user's login sessions are authenticated separately. If no terminal is present, the behavior is the same as ppid. Commands run from the same terminal will not require a password for dzdo.timestamp_timeout minutes (5 by default).

For example:

```
dzdo.timestamp_type:ppid
```

You can also set this parameter using group policy.

dzdo.tty_tickets

This configuration parameter specifies whether dzdo should require authentication once per-tty rather than once per user. The parameter value can be true or false. The default parameter value is false.

For example:

dzdo.tty_tickets: false

This configuration parameter provides functionality equivalent to the tty_tickets flag for configuring the sudoers file and sudo operation.

You can also set this parameter using group policy.

dzdo.use_pty

The dzdo.use_pty configuration parameter specifies if dzdo will always run commands in pseudo-terminal, also called a pseudo-pty or pseudo-tty. If you enable this parameter, dzdo runs commands in a pseudo-terminal even if there's no input or output logging. Using this option would make it impossible for a malicious program to run under dzdo to fork a background process that retains information to the user's terminal device after the main program finishes.

By default, this parameter is false.

dzdo.use.realpath

This configuration parameter specifies whether dzdo uses command paths resolved by realpath when searching for commands. The default parameter value is false, meaning that realpath is not used.

When set to true, this parameter specifies that realpath is used to expand all symbolic links and resolve references to:

- /./
- /../
- extra / characters

You can also set this parameter using group policy.

dzdo.user.command.timeout

When set to true, this parameter specifies a timeout on the DZDO command line with a -T option. If the timeout expires before the command has exited, the command is terminated.

The default setting is false.

dzdo.validator

This configuration parameter specifies the full path to a script that is executed each time the dzdo command is run. The script is run synchronously under the user's Active Directory name.

The dzdo command always runs the /usr/share/centrifydc/sbin/dzcheck script before it executes the command specified. However, the distribution package does not include a dzcheck script.

You do not need to create a dzcheck script to use dzdo. You only need to create a script if you want to modify dzdo behavior—for example, to prompt the user to enter some information before executing the command. To incorporate your modification, you would write the script, name it dzcheck and put it in /usr/share/centrifydc/sbin.

Use the dzdo.validator command only if you need to specify a different path or file name. (If you name your script dzcheck and store it at the default location, you do not need to use dzdo.validator.) For example, if the script was named myvalidator and it was in the /etc/centrifydc directory, you would add the following command in centrifydc.conf:

dzdo.validator: /etc/centrifydc/myvalidator

The dzdo command sets three environment variables:

- DZDO_USER: the Active Directory name of the user invoking dzdo
- DZDO_COMMAND: the command
- DZDO_RUNASUSER: the user name that the command will be run as

The script should return one of the following values:

- 0 Success. dzdo will continue and run the command.
- non-zero Failure: dzdo will not run the command. In this event, dzdo does NOT show a message on the console. If you want to notify the user of the failure, include the message in the script.

When the logging level is set to DEBUG, the call to the script and the return value are logged in var/log/centrifydc.log. If DEBUG is off, the call to the script and return value are logged in /val/log/messages.

dzdo.validator.required

This configuration parameter specifies whether dzdo is required to run the validator script. The default value is false.

> **Note**: The dzdo command skips the validator script if the script is not available, is not owned by root, or is group/world writable. By default, dzdo continues to run the command even if the validator script is skipped. When this parameter is set to true, dzdo does not run the command if validator script is skipped.

dzsh.roleswitch.silent

This configuration parameter specifies whether to display role information when changing from one role to another is a restricted shell.

By default, changing from one role to another displays a message indicating that you have changed your current role. For most commands that run in a restricted shell, displaying this message has no affect on the execution of the command.

There are cases, however, where a command—such as `sftp` or git—expects a specific type of response. Because the role change message is not the expected response, the message can cause the command to fail.

You can use this parameter to address those cases where the role change message would cause a command to fail.

Set this configuration parameter to true to prevent the role switch information from being displayed when running commands in a restricted shell. The default value is false.

For example:

dzsh.roleswitch.silent: true

Customizing Auto Zone Configuration Parameters

This section describes the configuration parameters that affect the operation of a local host computer joined to Auto Zone. These parameters have no effect if the computer is not joined to Auto Zone.

auto.schema.allow.groups

This configuration parameter specifies a list of Active Directory groups that define which Active Directory users are valid users in the Auto Zone. Members of the specified groups are considered valid users in the Auto Zone.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

Adding zone users based on group membership

By default, all Active Directory users are included in the Auto Zone. If you specify one or more groups using this parameter the only users who can log in using their Active Directory account are members of the specified groups, members of nested groups, users whose primary group is set to one of the groups specified, and all users specified in auto.schema.allow.users.

For example, to specify that only the members of the sf-adms and sf-apps groups should be allowed to log on to computers in Auto Zone, you would enter the following:

auto.schema.allow.groups: sf-adms sf-apps

The groups you specify for the auto.schema.allow.groups parameter must be security groups, but can be domain local, global, or universal groups.Distribution groups are not supported.

You can separate each group by a space or a comma and you can use double quotes or escape characters to include spaces or special characters in group names. For example: in group names. For example,

auto.schema.allow.groups: server_users, "Domain Admins", Domain\ Users

You should note that this parameter does not add the Active Directory groups you specify to Auto Zone. It does not assign the groups a numeric identifier (GID) or make the groups available to use as a primary group for any of the users added to Auto Zone. This parameter simply enables UNIX profiles for the users that are members of the specified groups. You can use the auto.schema.groups parameter to specify the Active Directory groups to include an in the Auto Zone and assign it a GID. You can configure the primary group for users using the auto.schema.primary.gid parameter.

Supported group name formats

You can specify groups by name or you can list the group names in a file using any of the following formats:

- SAM account name: sAMAccountName@domain
- User Principal Name: name@domain
- NTLM: DOMAIN/sAMAccountName
- Full DN: CN=commonName,...,DC=domain_component,DC=domain_component
- Canonical Name: domain/container/cn

The adclient process writes any group name that is not recognized to the agent log file.

Specifying the parameter value in a separate file

To specify a file that contains a list of Active Directory group names, you can set the parameter value using the file: keyword and a file location. For example:

auto.schema.allow.groups: file:/etc/centrifydc/auto_user_groups.allow

In the /etc/centrifydc/auto_user_groups.allow file, you would type each group name on its own line using any of the supported name formats. For example:

server_users
"Domain Admins"
Domain Users
CN=group6,CN=Users,DC=domain,DC=com

Limitations of this parameter

Auto Zone does not support one-way trusts. If there are any users in a specified group who belong to a domain that has a one-way trust relationship to the joined domain, they will not become valid users on the computer.

If you set this parameter, you should be aware of search limit defined for the auto.schema.search.return.max parameter. The setting for that parameter will limit the number of users returned in search results and stored in the cache. For example, if the auto.schema.search.return.max parameter is set to 100, and you use this parameter to specify an Active Directory group with 200 members, a query would only return results for the first 100 users. The remaining members of the group will still be allowed to log on to computers in the Auto Zone, but the results of queries might be misleading.

If desired, you can disable the auto.schema.search.return.max parameter by setting the parameter value to 0. Disabling the search limit ensures that all of the users in the specified Active Directory groups are listed as valid zone users when you run queries whether the number of users exceeds or falls short of the number specified for the auto.schema.search.return.max parameter. If you are not concerned about whether search results accurately reflect the users in the Active Directory groups you have defined for the auto.schema.allow.groups parameter, however, you don't need to modify the auto.schema.search.return.max parameter.

auto.schema.allow.users

This configuration parameter specifies which Active Directory users to include in the Auto Zone.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

Adding specific Active Directory users to Auto Zone

By default, all Active Directory users in a forest are included in the Auto Zone. If you specify one or more users using this parameter, however, only the specified users and members of the groups specified in the auto.schema.allow.groups parameter can log in using their Active Directory account.

For example, to specify that only the users jane and sai.wu should be allowed to log on to computers in Auto Zone:

auto.schema.allow.users: jane.doe sai.wu@ajax.org

You can separate each user name by a space or comma and use double quotes or escape characters to include spaces or special characters in group names. For example,

auto.schema.allow.groups: jane.doe, "Alex Adams", jae\ chin

Supported user name formats

You can specify users by name or you can list the user names in a file in any of the following formats:

- SAM account name: sAMAccountName@domain
- User Principal Name: name@domain
- NTLM: DOMAIN/sAMAccountName
- Full DN: CN=commonName,...,DC=domain_component,DC=domain_component
- Canonical Name: domain/container/cn

The adclient process writes any user name that is not recognized to the agent log file.

Specifying the parameter value in a separate file

To specify a file that contains a list of Active Directory user names, you can set the parameter value using the file: keyword and a file location. For example:

auto.schema.allow.users: file:/etc/centrifydc/auto_user_users.allow

In the /etc/centrifydc/auto_user_users.allow file, you would type each user name on its own line using any of the supported name formats. For example:

jane.doe
sai/wu@ajax.org
CN=Alex Adams,CN=Users,DC=ajax,DC=org

auto.schema.apple_scheme

This configuration parameter specifies that you want to use the Apple algorithm to automatically generate user and group identifiers. The Apple algorithm for generating identifiers is based on the objectGuid attribute for the user or group object. The Delinea mechanism for automatically generating UIDs and GIDs is based on the security identifier for the user or group objects. Both methods ensure a globally unique and consistent identifier for the user or group.

By default, this parameter value is set to false. If you want to use the Apple algorithm, set the parameter value to true. For example:

auto.schema.apple_scheme: true

If you set this parameter to use the Apple algorithm, you must use adflush to clear the cache, then restart the adclient process to update UIDs, GIDs, and user primary GIDs. Note that the user's primary group must be available in Auto Zone. If a user's primary group is not in the zone, the user will have an incomplete profile and unable to log on. If a user is provisioned with an incomplete profile, an error is recorded in the Window Event log.

After clearing the cache and restarting the agent, run the fixhome.pl script to correct conflicts between the new user UID and the home directory ownership.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

auto.schema.domain.prefix

This configuration parameter specifies a unique prefix for a trusted domain. You must specify a whole number in the range of 0 to 511.

The Delinea algorithm for generating unique identifiers combines the prefix with the lower 22 bits of each user or group RID (relative identifier) to create unique UNIX user (UID) and group (GID) IDs for each user and group in the forest and in any two-way trusted forests.

Ordinarily, you do not need to set this parameter because the Delinea Agent automatically generates the domain prefix from the user or group security identifier (SID). However, in a forest with a large number of domains, domain prefix conflicts are possible. When you join a computer to a domain, the Centrify Agent checks for conflicting domain prefixes. If any conflicts are found, the join fails with a warning message. You can then set a unique prefix for the conflicting domains.

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

To set this parameter, append the domain name and specify a prefix in the range 0 - 511. For example:

auto.schema.domain.prefix.acme.com: 3
auto.schema.domain.prefix.finance.com: 4
auto.schema.domain.prefix.corp.com: 5

The default behavior, if you do not set this parameter, is for the agent to automatically generate the domain prefix from the user or group security identifier (SID).

auto.schema.groups

This configuration parameter specifies the Active Directory groups to include in the Auto Zone. When you specify one or more groups in this parameter, the groups specified are assigned a group ID on this computer.

The command syntax is:

auto.schema.groups: groupname [, groupname, groupname, ...]

By default all Active Directory groups are included.

> **Note**: If an Active Directory user specified in auto.schema.allow.users is a member of a group and that group is NOT specified in auto.schema.groups, that group is ignored.

Any groups listed under auto.schema.groups can be domain local, global or universal security groups. Distribution groups are not supported.

You specify each group by name or you can list the groups in a file. The group name can be specified in any of the following formats:

- SAM account name: sAMAccountName@domain

  (specify the domain if the group is not in the current domain)

- User Principal Name: name@domain
- NTLM: DOMAIN/sAMAccountName

  > **Note**: Use the adclient.ntlm.separators parameter to specify different NTLM separators.

- Full DN: CN=commonName,...,DC=domain_component,DC=domain_component
- Canonical Name: domain/container/cn

adclient writes any name that is not recognized to the agent log file.

You can also define the groups using group policy.

Examples:

auto.schema.groups: finance_users
auto.schema.groups: “Mktg Users”
auto.schema.groups: ops@domain.com

You can specify multiple groups in a single command. Separate each group by a comma and use escape characters to include, for example, spaces, backslashes, or a comma in the group specification. For example,

auto.schema.allow.groups: server_users, "Domain Admins", Domain\ Users, \ group1, group2@domain.com, domain\\group3, domain+group4, \
domain/group5, CN=group6\,CN=Users\,DC=domain\,DC=com, \
domain/Users/group7

You can also use a file instead. The syntax is file:/path. For example,

auto.schema.allow.groups: file:/etc/centrifydc/auto_user_groups.allow

In the file, enter each group line by line. However, you do not need the escape characters. For example, the following list enters the same groups as the previous example:

server_users
"Domain Admins"
Domain Users
group1
group2@domain.com
domain\group3
domain+group4
domain/group5

CN=group6,CN=Users,DC=domain,DC=com
domain/Users/group7

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

auto.schema.homedir

This configuration parameter specifies the home directory for logged in users. The default, if you do not specify this parameter, is:

- Mac OS X: /Users/%.
- Linux, HP-UX, AIX: /home/%
- Solaris: /export/home/%

The syntax % specifies the logon name of the user. For example, in the centrifydc.conf configuration file, if you add:

auto.schema.homedir:/Users/%

and jsmith logs on to a Mac OS X machine, the home directory is set to /Users/jsmith.

If the parameter auto.schema.use.adhomedir is true, the home directory is set to the value in Active Directory for the user, if one is defined. If auto.schema.use.adhomedir, is false or if a home directory is not specified for the user in Active Directory, the home directory is set to the value defined for this parameter, auto.schema.homedir.

> **Note**: The configuration parameter auto.schema.homedir.illegal_chars defines characters that are not allowed in home directory names. Any illegal characters in the specified name are removed from the home directory name on the computer.

You can also specify the home directory on all machines joined to Auto Zone by using group policy.

auto.schema.primary.gid

This configuration parameter specifies the primary GID for Auto Zone users. The auto.schema.private.group parameter must be set to false to use this parameter.

> **Note**: On Mac OS X, the default value of auto.schema.private.group is false. On Linux, HP-UX, Solaris, and AIX, the default value of auto.schema.private.group is true.

To specify the GID for an existing group, you must first find the GID for the group. To find the GID for a group, you can use the adquery command or adedit. For example, to find the GID for the group Support, open a terminal session and type:

```
>adquery group --gid Support
```

The command returns the GID for the Active Directory group Support:

```
1003
```

You can then set this parameter to the value returned. For example:

```
auto.schema.primary.gid: 1003
```

If you do not set this parameter, the value defaults to the following:

- On Mac OS X:

  ```
  auto.schema.primary.gid: 20
  ```

- On Linux, HP-UX, Solaris, and AIX:

  ```
  auto.schema.primary.gid: -1
  ```

If you are using the Apple algorithm to automatically generate user and group identifiers, including the group identifier for primary groups, set this parameter to -1 to disable it. For example:

```
auto.schema.primary.gid: -1
```

In most cases, you set this configuration parameter using group policy. You can, however, set it manually in the configuration file if you are not using group policy or want to temporarily override group policy.

auto.schema.private.group

This configuration parameter specifies whether to use dynamic private groups.

Specify true to create dynamic private groups. If you specify true, the primary GID is set to the user's UID and a group is automatically created with a single member.

Specify false to not create private groups.

On Mac OS X, the default value of this parameter is false. On Linux, HP-UX, Solaris, and AIX, the default value is true.

If you specify false, the primary GID is set to the value of auto.schema.primary.gid. On Mac OS X, the default value of auto.schema.primary.gid is 20. On Linux, HP-UX, Solaris, and AIX, the default value of auto.schema.primary.gid is -1.

auto.schema.shell

This configuration parameter specifies the default shell for the logged in user. The default value is:

- /bin/bash on Mac OS X and Linux systems
- /bin/sh on UNIX systems, including Solaris, HPUX, AIX.

You can also set the default shell on all machines joined to Auto Zone by using group policy.

auto.schema.use.adhomedir

Note: This configuration parameter applies to Mac OS X computers only.

This configuration parameter specifies whether or not to use the Active Directory value for the home directory if one is defined. Set to true to use the Active Directory value (the default), or false to not use the Active Directory value. If you set the value to false, or if you set the value to true but a home directory is not specified in Active Directory, the value for auto.schema.homedir is used.

auto.schema.name.format

This configuration parameter specifies how the Active Directory username is transformed into a UNIX name (short name in Mac OS X). The options are

- SAM (default)

  An example SAM name is joe

- SAM@domainName

  An example SAM@domainName is joe@acme.com

- NTLM

  An example NTLM name is acme.com-joe

auto.schema.separator

> **Note**: This configuration parameter has been deprecated in favor of adclient.ntlm.separators, which applies whenever NTLM format is used. The auto.schema.separator parameter only applies when the computer is connected to Auto Zone.

This configuration parameter specifies the separator to be used between the domain name and the user name if NTLM format is used. The default separator is a plus (+). For example:

auto.schema.separator:+

which results in a name such as:

acme.com+jcool

auto.schema.search.return.max

This configuration parameter specifies the number of users that will be returned for searches by utilities such as dscl and the Workgroup Manager application. Because Auto Zone enables access to all users in a domain, a search could potentially return tens of thousands of users. This parameter causes the search to truncate after the specified number of users.

The default is 1000 entries.

auto.schema.name.lower

This configuration parameter converts all usernames and home directory names to lower case in Active Directory.

Set to true to convert usernames and home directory names to lowercase.

Set to false to leave usernames and home directories in their original case, upper, lower, or mixed.

The default for a new installation is true.

auto.schema.iterate.cache

This configuration parameter specifies that user and group iteration take place only over cached users and groups.

Set the value for auto.schema.iterate.cache to true to restrict iteration to cached users and groups.

Set the value for auto.schema.iterate.cache to false to iterate over all users and groups. The default value is false.

auto.schema.uid.conflict

This configuration parameter specifies what is done if adclient discovers that an Active Directory user already exists with the same UID. There are two options:

- allow: Allow the duplicate UID; an information message is logged.
- disallow: If a duplicate UID already exists, the second user with the same UID is ignored; a warning message is logged.

Examples:

auto.schema.uid.conflict: allow
auto.schema.uid.conflict: disallow

auto.schema.homedir.illegal_chars

This configuration parameter specifies the characters in a home directory name that are not allowed in UNIX, Linux or Mac OS X home directory names. Each character in a home directory name that matches one of the specified characters is simply removed from the name; for example:

/home/user$34 /* illegal $ character
/home/user34 /* illegal character removed

The default setting in centrifydc.conf for UNIX (HP-UX, Solaris, AIX) and Linux systems is the following:

auto.schema.homedir.illegal_chars: \t\n /\\$><?*%|\"\'`[]

The default setting in centrifydc.conf for Mac OS X systems is the following (space is omitted):

auto.schema.homedir.illegal_chars: \t\n/\\$><?*%|\"\'`[]

Run the adflush command after you change the value to flush the cache.

auto.schema.thycotic.rids

Use this parameter to specify that the service generates UIDs and GIDs based on the algorithm used by Delinea Secret Server instead of the algorithm that Server Suite uses.

By default, this parameter is set to false, which means that the service generates UIDs and GIDs based on Server Suite.

The Delinea UID/GID generation algorithm is as follows:

uid/gid = RID of object * 100 + suffix

The service generates the suffix by getting a list of the Active Directory domain names that are in the Active Directory forest and then sorts that list. The suffix is the same as the index, but it starts from 1 instead of zero.

auto.schema.unix.name.disallow.chars

This configuration parameter specifies the characters in an Active Directory user or group name that are not allowed in UNIX, Linux or Mac OS X names. Each character in the name that matches the characters specified is replaced in the corresponding UNIX name by the character specified in auto.schema.substitute.chars.

> **Note**: Be sure to specify the replacement character in auto.schema.substitute.chars. Otherwise, the offending character is simply removed from the name, and you run the risk of duplicate UNIX names.

The default setting in centrifydc.conf for UNIX (HP-UX, Solaris, AIX) and Linux systems is the following:

auto.schema.unix.name.disallow.chars: \t\n /\\><?|\"\'`[],:;~!@#$%^&*()=

The default setting in centrifydc.conf for Mac OS X systems is the following (space is omitted):

auto.schema.unix.name.disallow.chars: \t\n/\\><?|\"\'`[],:;~!@#$%^&*()=

Run the adflush command after you change the value to flush the cache.

auto.schema.substitute.chars

This configuration parameter specifies the character that replaces any characters specified in auto.schema.unix.name.disallow.chars encountered in an Active Directory user or group name in the corresponding UNIX name.

The default setting in centrifydc.conf is the following:

auto.schema.substitute.chars: _ (underbar)

Run the adflush command after you change the value to flush the cache.

auto.schema.max.unix.name.length

This configuration parameter specifies the maximum length for a generated UNIX user or group name. The UNIX names are generated from the Active Directory user and group names.

The default setting in centrifydc.conf is the following:

auto.schema.max.unix.name.length: 33

Run the adflush command after you change the value to flush the cache.

Customizing Auditing Configuration Parameters

This section describes the configuration parameters that you can set on audited computers that have a Centrify Agent for *NIX installed. These parameters affect the operation of the auditing service (dad) and the UNIX shell wrapper (cdash).

The parameters are defined in a text file named centrifyda.conf in /etc/centrifyda on each audited computer.

> **Note**: For information about specifying an audit trail target in the centrifydc.conf file, see audittrail.targets.

agent.max.missed.update.tolerance

This configuration parameter specifies the number of unsuccessful attempts the Delinea auditing agent makes to join a collector before displaying a notification that the agent is not joined to a collector. Each attempt is made after an interval of 5 minutes.

For example, if you want the agent to warn you that it is not connected to a collector after 3 attempts, you would enter the following:

agent.max.missed.update.tolerance: 3

The default value for this parameter is 4.

You can use this parameter with the dad.collector.connect.timeout parameter which specifies the amount of time, in seconds, the agent waits during each connection attempt before it determines that it cannot connect to a collector.

agent.send.hostname

This configuration parameter enables audited sessions to display the host name specified by the agent on audited computers instead of the host name resolved by the collector through DNS. This configuration parameter is useful in configurations where the DNS servers used by the collectors cannot reliably resolve host names from IP addresses. The most common scenarios that might require you to use this configuration parameter are when the agents are in a virtual environment using network address translation (NAT) or in a perimeter network outside of a firewall.

You can set this parameter to true if you want the agent to determine the host name used. If you set this configuration parameter to false, the collector determines the agent's host name based on its IP address.

For example:

agent.send.hostname: true

agent.video.capture

This configuration parameter enables or disables saving the video capture for a specific agent, which overrides the video capture settings configured for the entire DirectAudit installation. If video capture is disabled, the collector does not display the video output and does not save it to the database.

The default value uses the settings that you have configured for the entire installation.

To save the captured video output to the database, you set this parameter to enabled. To not save the captured video output, set this parameter to disabled.

For example:

agent.video.capture: enabled

autofix.nss.conf

This configuration parameter enables dad to fix /etc/nsswitch.conf automatically if anything goes wrong. If set to true, the dad process configures the /etc/nsswitch.conf file automatically. If set to false, the /etc/nsswitch.conf file is left unmodified.

The default value is true.

For example, to disable changes to the /etc/nsswitch.conf file:

autofix.nss.conf: false

cache.enable

This configuration parameter controls whether the dad process caches name service query results about users and groups. If set to true, the dad process stores query results, for example, from user lookup requests, in memory for better performance. If set to false, query results are not saved and must be retrieved whenever they are needed.

If set to true, you can use the cache.max.size and cache.time.to.live parameters to control the number and duration of entries in the cache. You can also use the daflush command to clear the cache manually when you want to ensure you get updated information. For example, if you remove the UNIX Login role for an Active Directory user, some information for that user might remain in the cache and be returned when you run a command such as getent passwd. You can run daflush to ensure the user is removed completely from the local computer cache, including the auditing name service cache.

The default value for this parameter is true.

For example, to disable the name service cache on an audited computer:

cache.enable: true

cache.max.size

This configuration parameter controls the maximum number of entries that can be stored in the dad name service cache. query results about users and groups. This parameter is only applicable if the cache.enable parameter is set to true. The dad process stores query results up to the value set for this parameter in memory for better performance.

The default value for this parameter is 80,000 entries.

For example, to increase the maximum number of name service results that can be stored on an audited computer:

cache.max.size: 85000

cache.time.to.live

This configuration parameter controls the length of time entries should remain valid in the name service cache. You can specify the maximum number of seconds cached query result should be available in the cache. This parameter is only applicable if the cache.enable parameter is set to true.

The default value for this parameter is 10 minutes (600 seconds).

For example, to increase the number of seconds query results are available in the cache on an audited computer:

cache.time.to.live: 900

cagent.audit.session

This configuration parameter determines if session auditing is enabled with the Delinea Client. A value of 1 means that session auditing is enabled and a value of 0 (zero) means that session auditing is disabled. The default value is 1.

To change the setting, run dacontrol -d to disable auditing or dacontrol -e to enable auditing. Do not use cedit to edit this parameter.

dad.client.idle.timeout

This configuration parameter determines the time interval within which dad checks for disconnected dash sessions.

The default value for this parameter is 30 minutes (1800 seconds).

dad.collector.connect.timeout

This configuration parameter specifies the amount of time, in seconds, the agent waits during each connection attempt before it determines that it cannot connect to a collector.

The default value for this parameter is 60 seconds.

You can use this parameter with the agent.max.missed.update.tolerance parameter which allows you to specify the number of unsuccessful attempts that the agent can make to connect to a collector before notifying the user that it is not connected to a collector.

dad.dumpcore

This configuration parameter enables dad to do a core dump if an audited computer crashes. This parameter overrides the default ulimit setting. If set to true, the agent will generate a core dump if the computer crashes. If set to false, no core dump is generated.

The default value is **false.**

dad.gssapi.seal

This configuration parameter specifies whether the auditing service seals network communications with the collector using a secure GSSAPI connection. If set to **true**, the network connection is sealed and cannot be read. If set to **false**, the connection is not sealed and is humanreadable.

The default value is **true.**

dad.gssapi.sign

This configuration parameter specifies whether the auditing service signs network communications with the collector over a secure GSSAPI connection. If set to **true**_, the network connection is signed. If set to **false**, the connection is not signed.

The default value is **true.**

dad.process.fdlimit

This configuration parameter specifies the number of file descriptors that can be used for audited sessions.

For some UNIX platforms, such as Solaris, the default number of available file descriptors for each process is insufficient of auditing sessions, because the Delinea Agent requires two descriptors per session.

Use this parameter to increase the number of file descriptors available. For example:

dad.process.fdlimit: 2048

This configuration parameter can also be set using Group Policy.

dad.resource.cpulimit

This configuration parameter specifies the maximum percentage of CPU usage that dad can use before dad is restarted, or before the event is logged in /var/log/centrifydc.log. Whether dad is restarted when the threshold is exceeded is controlled by [dad.resource.restart]dad-resource-restart.md).

The default value of this parameter is 50, meaning that dad is restarted or the event is logged when dad CPU usage exceeds 50%. For example:

dad.resource.cpulimit: 50

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset. See dad.resource.restart for more information about the advantages of setting a threshold that is lower than the default system value.

dad.resource.cpulimit.tolerance

This configuration parameter specifies the number of times that CPU usage can exceed the threshold set in dad.resource.cpulimit before dad is restarted, or before the event is logged in /var/log/centrifydc.log. Whether dad is restarted when the value of this parameter is exceeded is controlled by dad.resource.restart.

The default value of this parameter is 5, meaning that the CPU usage set in dad.resource.cpulimit can be exceeded four times before dad is restarted on the fifth instance or the event is logged. For example:

dad.resource.cpulimit.tolerance: 5

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset. See dad.resource.restart for more information about the advantages of setting a threshold that is lower than the default system value.

dad.resource.fdlimit

This configuration parameter specifies the maximum file descriptors that dad can use before dad is restarted, or before the event is logged in /var/log/centrifydc.log. Whether dad is restarted when the threshold is exceeded is controlled by dad.resource.restart.

The default value of this parameter is 500, meaning that dad is restarted or the event is logged when dad file descriptor usage exceeds 500. For example:

dad.resource.fdlimit: 500

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset. See dad.resource.restart for more information about the advantages of setting a threshold that is lower than the default system value.

dad.resource.memlimit

This configuration parameter specifies the maximum memory usage (in bytes) that dad can use before dad is restarted, or before the event is logged in /var/log/centrifydc.log. Whether dad is restarted when the threshold is exceeded is controlled by dad.resource.restart.

The default value of this parameter is 104857600 (100 MB), meaning that dad is restarted or the event is logged when dad memory usage exceeds that value. For example:

dad.resource.memlimit: 104857600

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset. See dad.resource.restart for more information about the advantages of setting a threshold that is lower than the default system value.

dad.resource.restart

This configuration parameter specifies whether dad restarts or just logs the event when a resource threshold is exceeded. Events that are logged are recorded in /var/log/centrifydc.log.

The default value of this parameter is false, meaning that when a resource threshold is exceeded, the event is logged, but dad is not restarted. For example:

dad.resource.restart: false

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

You can set resource thresholds in these parameters:

- dad.resource.cpulimit
- dad.resource.cpulimit.tolerance
- dad.resource.fdlimit
- dad.resource.memlimit
- dad.resource.timer

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset.

Setting a low threshold for restarting dad and purging the client can avoid problems with resources being consumed prematurely. For example, when cdash calls another cdash recursively, dad receives a large number of client requests. Solaris has only 256 file descriptors for ulimit by default. Unless you configure a threshold lower than 256, cdash does not stop recursive calling until 257 calls, and all of dad's file descriptors could be consumed by that one operation.

dad.resource.timer

This configuration parameter specifies how often (in seconds) the dad resource monitor checks dad resource usage.

The default value of this parameter is 600 (10 minutes), meaning that the dad monitor checks dad resource usage every 10 minutes. For example:

dad.resource.timer: 600

The dad resource monitor automatically checks the usage of various dad resources during runtime. For each resource that is monitored, you can configure the threshold value that triggers a dad restart or a log entry.

When dad is restarted, the client is purged, and counters for resources such as CPU usage, file descriptors, and memory are reset. See dad.resource.restart for more information about the advantages of setting a threshold that is lower than the default system value.

dad.timer.diskspace

This configuration parameter specifies the number of seconds between checks of disk space when the disk space reserved for offline storage is less than the value specified in the spool.diskspace.min parameter.At each check, a warning message is written to the log file.

The default value is 360 seconds.

dad.timer.monitor.nss.conf

This configuration parameter controls how frequently the dad process checks the /etc/nsswitch.conf file for changes. Set this parameter to the number of seconds between checks.

The default value is 60 seconds.

dash.allinvoked

This configuration parameter was previously required to support auditing of shells invoked in scripts and command-level auditing. The parameter is no longer required and can be removed if you upgrade the agent to the latest version and enable command-level auditing through NSS. If you do not update the agent, you can use this parameter to specify whether to audit all shell invocations. If set to **true**, all login and non-login shells are audited. If set to **false**, only login shells and login sub-shells are audited. If set to false, invoked shells are not audited.

The default value is **false**.

dash.auditstdin

This configuration parameter specifies whether the agent captures standard input (stdin). If set to true, the auditing service records all session input and output, including stdin data. If set to false, the auditing service records all session activity to standard output, but does not capture stdin data.

The default value is true.

dash.auditstdin.except

This configuration parameter specifies strings that cdash should ignore when capturing stdin data. For security, typed passwords are always ignored by default. Use regular expressions and do not include quotes. Leading and trailing spaces are ignored, spaces in the middle are not affected. For example:

dash.auditstdin.except: (prompt1|prompt2)

will match strings like these:

This is prompt1:
Prompt2 asks for password:

The default value is empty to only ignore the passwords that users enter. For more information about specifying exceptions, see the comments in the centrifyda.conf file.

dash.cmd.audit.blacklist

This configuration parameter specifies whether certain privileged command patterns are skipped while auditing is enabled.

To add a command pattern to skip, list the regular expression you wish to skip to this parameter. When enabled, an audit trail is still sent by the agent, but the specified command and arguments will not be captured. For example, the following list will skip auditing of the “date” command:

dash.cmd.audit.blacklist: date

By default, no command patters are skipped while auditing.

dash.cmd.audit.show.actual.user

This configuration parameter specifies whether command-based auditing records will display the actual user account used to run a privileged command that requires auditing, as well as the run-as account.

By default, the value of this parameter is set to false, and only the run-as account used to execute privileged commands is shown in auditing records. To enable this parameter, set the value to true. For example:

dash.cmd.audit.show.actual.user: true

dash.cont.without.dad

This configuration parameter specifies whether cdash prompts the user to restart auditing when it determines that dad is not running. If set to true, cdash does not prompt the user to restart auditing and continues without the dad process. If set to false, cdash prompts the user to restart auditing to continue.

The default value is false.

dash.force.audit

This configuration parameter specifies one or more session binary files to audit. This parameter was previously required to support command-level auditing on managed computers. The parameter is no longer required and can be removed if you upgrade the agent to the latest version.

Instead of setting this parameter, you must run the following command to enable auditing for specific command-line programs:

dacontrol --enable --command cmd_path

If you do not update the agent, you can use this parameter to specify commands to be audited by appending .daudit to the file name. For example, to audit secure shell (ssh) sessions:

dash.force.audit: /usr/share/centrifydc/bin/ssh.daudit

However, you still must run the dacontrol command to enable auditing for specific commands.

You can separate entries by typing a space or a comma. You can escape spaces or commas in file names using the backslash character (\). This parameter is not included in the configuration file by default.

dash.loginrecord

This configuration parameter specifies whether the auditing service should add utmp entries for the cdash pseudo terminals (pty). The setting of this parameter affects the results of whoami and who commands.

If set to **true**, the auditing service adds utmp entries for cdash pseudo terminals processes. With this setting, whoami in an audited shell works as expected, but who commands list logged-in users twice.

If set to false, the auditing service does not create additional utmp entries. With this setting, the whoami command in an audited shell cannot determine complete user information. Workaround: on some operating systems: who --lookup works, but the who command lists users only once.

The default value is false.

dash.obfuscate.pattern

This configuration parameter enables you to hide sensitive information in the standard output (stdout) in audit results by using patterns to define the hidden information.

Beginning with release 2015.1, each pattern that you create must be embedded inside double quote characters (“). For example:

“nnnn-nnnn-nnnn-nnnn”

> **Note**: If there is a delay in the display of standard output information that you have designated to be hidden by defining a pattern, the agent may not recognize the pattern, and the information may be shown. To avoid delays, your obfuscation string should not exceed the size of the Cdash standard output buffer, which is 4KB.
>
> **Note**: In releases earlier than 2015.1, patterns could be embedded inside double quote characters (“) or slash characters (/). If slash characters were used, they are converted automatically to double quote characters when you upgrade from 2015 to 2015.1.

Each single character in a pattern corresponds to one character in actual session data.

Supported characters in a pattern are as follows:

A	Any upper case letter.
d	Any character.
D	Any letter.
n	Any decimal digit character.
s	Symbols, such as the following: ~ ` ! @ # $ % ^ & * (- _ = + [{] } | \ : ; ' < , > . ? /
-	Separator for exact matching in session data.
_	Separator for exact matching in session data.
(	Separator for exact matching in session data.
)	Separator for exact matching in session data.
,	Separator for exact matching in session data.
.	Separator for exact matching in session data.

If you define more than one pattern, separate the patterns with spaces. For example:

“nnnn-nnnn” “A-nnnn”

By default, this parameter does not contain any patterns.

dash.obfuscate.regex

This configuration parameter enables you to hide sensitive information in the standard output (stdout) in audit results by using regular expressions to define the hidden information patterns.

Beginning with release 2015.1, each regular expression that you create must be embedded inside double quote characters (“). For example:

“[A-Z][0-9]{6}\\([0-9A-Z]\\)”

If you define more than one regular expression, separate the regular expressions with spaces. For example:

“[0-9]-[0-9]” “[a-z]-[0-9]”

> **Note**: If there is a delay in the display of standard output information that you have designated to be hidden by defining a pattern, the agent may not recognize the pattern, and the information may be shown. To avoid delays, your obfuscation string should not exceed the size of the Cdash standard output buffer, which is 4KB.
>
> **Note**: In releases earlier than 2015.1, patterns could be embedded inside double quote characters (“) or slash characters (/). If slash characters were used, they are converted automatically to double quote characters when you upgrade from 2015 to 2015.1.

By default, this parameter does not contain any regular expressions.

dash.obfuscate.stdin

This configuration parameter specifies whether cdash hides (obfuscates) the STDIN. Setting this parameter to true is useful in situations where users enter sensitive data such as login credentials and so forth. The service hides the STDIN data according to the patterns specified in the dash.obfuscate.pattern and dash.obfuscate.regex parameters.

> **Note**: If the user uses line-editing keys such as left arrow, right-arrow, and so forth on the command line, the obfuscation will fail. The exception is that if a user uses the backspace key, the obfuscation will still occur.

By default, this parameter is set to false.

dash.parent.skiplist

This configuration parameter lists the names of parent processes that should not be audited. If the name of a process's parent is in this list, cdash will drop out without auditing.

You can add parent processes to the list or remove the default parent processes if you do not want to skip auditing for these processes. List entries must be separated by spaces.

For example, to skip auditing for the sapstartsrv, gdm-binary, gdm-session-wor, kdm, and sdt_shell parent processes:

dash.parent.skiplist: sapstartsrv gdm-binary gdm-session-wor kdm sdt_shell

You can also set this parameter using group policy.

dash.prompt.message.file

This parameter specifies a file that contains auditing messages that display to the user in various situations. The default location of the file is /etc/centrifda/dash.prompt.message. In that file are some key-value pairs that you can customize to your preferred wording or language.

The following situations apply if the user is set as "Audit required" but also doesn't have the "Always permit logon" right:

- dad.stopped.purposely - If an administrator intentionally stops the auditing service (also referred to as the DirectAudit daemon, or dad), the following message displays:

- dad.stopped.unexpectedly - If the auditing service stops unexpectedly, the following message displays:

- dad.diskspace.too.low- If there isn't enough disk space on the audited system, the following message displays:

fails, contact your administrator to increase the available disk space and then try again.

dash.reconnect.dad.retry.count

This configuration parameter specifies how many times cdash attempts to connect to dad after dad has started.

The default value is 3 retry attempts.

dash.reconnect.dad.wait.time

This configuration parameter specifies the number of seconds to wait after restarting dad before attempting to reconnect to dad.

The default value is 1 second.

dash.select.timeout

Use this parameter to specify the number of milliseconds that cdash waits while checking if the data is ready.

The default value is -1, which means that cdash determines the timeout dynamically.

> **Note**: If there is no data ready to be obfuscated in the timeout period, cdash flushes the buffered data to the collector immediately. Because of this, it's recommended that you specify a timeout value that is larger than the delay between keystrokes: for example, 5000 (equal to 5 seconds). Otherwise, cdash might split the data into multiple segments which would not match the patterns specified in the dash.obfuscate.pattern and dash.obfuscate.regex parameters and therefore the data obfuscation fails.

dash.shell.env.var.set

This configuration parameter specifies whether cdash should set the SHELL environment variable to the user's actual shell or the audit shell.

If set to true, the SHELL environment variable is set to the user's actual shell. If set to false, the SHELL environment variable is set to the audited shell.

The default is true.

dash.ssh.command.skiplist

This configuration parameter specifies the commands that can be executed using a secure shell (ssh) connection without being audited. You can use this parameter to prevent the auditing service from capturing unwanted session information. For example, by setting this parameter, you can avoid recording all of the binary data sent to and from the server when you execute file transfer commands such as rsync, sftp, or scp through a secure shell connection. By default, the parameter is configured to skip auditing for the rsync, sftp and scp commands, which are the most commonly used file transfer programs.

You can add programs to the list or remove the default programs if you don't want to skip auditing for these sessions. If you remove file transfer programs from the list, however, long data streams might cause problems when transferred to collector service.

For example, to skip auditing for ftp, rsync, sftp, scp, and wget commands:

dash.ssh.command.skiplist: ftp rsync sftp scp wget

dash.user.alwaysallowed.list

This configuration parameter lists the names of UNIX users who are allowed to use a session even if the computer cannot be audited due to environment setup issues.

By default, root is the only user allowed to use an unaudited session.

To use this parameter, specify a space-separated list of UNIX user names.

You can also set this parameter using group policy.

dash.user.skiplist

This configuration parameter lists the names of UNIX users and Active Directory users with a UNIX login who should not be audited. You can separate user names by typing a space or a comma. For example:

dash.user.skiplist: MaeJones kelly,dmorris,BookerJames

The default value is empty.

When you list a user in the dash.user.skiplist, this overrides the user's audit level. For example, if the user is set as "Audit Required" and also in dash.user.skiplist, the user might log in successfully but without being audited.

event.execution.monitor

Use the event.execution.monitor parameter to monitor all programs that users run in an audited session.

To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m.

The default value for the event.execution.monitor parameter is false.

In the audit.log file, you can find these events by looking for the cda_sys_execve messages. In the cdc.log file you can find them by looking for the Emit COMMAND_HISTORY.

event.execution.monitor.user.skiplist

Use the event.execution.monitor.user.skiplist parameter to specify a list of users to exclude from advanced monitoring for program execution. For these users, the auditing service does not record any programs that they run, even when the parameter event.execution.monitor is set to true.

For users specified in this list, the auditing service checks for the user account that the program is run by, also known as the “run as” user account.

To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m.

event.file.monitor

Use the event.file.monitor parameter to enable advanced monitoring for configuration files. To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m.

If advanced monitoring is enabled for files, the auditing service monitors any activity in the following folders:

- /etc/
- /var/centrify/
- /var/centrifydc/
- /var/centrifyda/

The default value for the event.file.monitor parameter is true.

In the audit.log file, you can find these events by looking for the cda_file_monitor_write messages. In the cdc.log file you can find them by looking for the Emit AUDIT_TRAIL.

event.file.monitor.process.skiplist

Use the event.file.monitor.process.skiplist parameter to specify which processes to not monitor that are in an area that you've already configured to provide detailed file monitoring. To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m.

When either adclient or dad processes or one of their sub-processes access or alter one of the files specified in the event.file.monitor list, this activity is automatically excluded from advanced monitoring.

The default value for this parameter is daspool.

event.file.monitor.user.skiplist

Use the event.file.monitor.user.skiplist parameter to specify a list of users to exclude from advanced monitoring for files. For these users, the auditing service does not record any write access to directories specified in event.file.monitor.

For users specified in this list, the auditing service checks this list against the original login user.

For example:

The event.file.monitor.user.skiplist parameter does not include the user dwirth. Dwirth uses the following command:

dzdo cp /tmp/badfile /etc/badfile

This activity generates the following audit event:

user dwirth run as root opened the file /etc/badfile using the /bin/cp command.

To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m.

The default value for this parameter is root.

event.monitor.commands

Use the event.monitor.commands parameter to specify a list of commands to monitor. Be sure to list each command with the full path. The auditing service generates an audit trail event when a user runs any of these monitored commands, and ignores any commands listed in the event.monitor.commands.user.skiplist.

To use this parameter, you must have enabled the agent to perform advanced monitoring with the command dacontrol -m. Otherwise, you will not get any report or audit trail event results.

In the audit.log file, you can find these events by looking for the cda_cmd_exec messages. In the cdc.log file you can find them by looking for the Emit AUDIT_TRAIL and Emit COMMAND_HISTORY messages.

event.monitor.commands.user.skiplist

Use the event.monitor.commands.user.skiplist parameter to specify any users for whom you do not want to monitor the commands that they run. Any user that you specify in this parameter will not generate an audit trail event when they run any command, even if the command is listed in event.monitor.commands.

lang_setting

This configuration parameter specifies the code page that is used by DirectAudit for character encoding.

You can set this parameter to one of the following valid values:

- utf8
- iso8859-1

The default value is utf8.

You can also set this configuration parameter using group policy.

lrpc2.message.signing

Use the lrpc2.message.signing to manage message singing behavior.

You can set this parameter to one of the following values:

- Disabled (default) to not ever sign LRPC2 messages.
- Allowed does not require LRPC2 message signing but may be chosen if required.
- Required must sign LRPC2 messages.

lrpc2.timeout

This configuration parameter specifies the number of seconds cdash and dainfo waits while trying to contact the dad service before timing out.

The default value is 30 seconds.

You can also set this configuration parameter using group policy.

lrpc2.rebind.timeout

This configuration parameter specifies the number of seconds that dareload (-b) waits while trying to connect to the dad service before timing out.

The default value is 300 seconds.

You can also set this configuration parameter using **for rebinding** group policy.

nss.alt.zone.auditlevel

This configuration parameter enables you to specify a default audit level for all users who do not have an audit level explicitly defined using the nss.user.override.userlist parameter. Note that this parameter is only applicable in classic zones. You should not set this parameter if you are using hierarchical zones.

You can set this parameter to one of the following valid values:

- use_sysrights
- audit_if_possible
- no_audit
- audit_required

The default value is use_sysrights. This setting determines the audit level by communicating with the adclient process.

The effective audit level for a user is determined in the following order:

1. If the user is included in the list specified by the nss.user.override.userlist parameter and the audit level is specified for the user, the specified audit level is used.
2. If the user is included in the list specified by the nss.user.override.userlist parameter and the audit level is not specified for the user, the value specified by the nss.user.override.auditlevel parameter is used.
3. If the user is not included in the list specified by the nss.user.override.userlist parameter, the audit level specified for the nss.alt.zone.auditlevel parameter is used.

nss.nologin.shell

This configuration parameter enables you to specify one or more non-login shells for audited users. In most cases, when audited users log on, they are placed in a wrapper shell so that their activity can be captured and sent to a collector. To use a real shell instead of the wrapper shell, you can specify shells to be non-login shells for audited users to access. After you set this parameter, you should restart the auditing service (dad).

For example, to define the shells /sbin/shell_test1 and /bin/shell_test2 as a non-login shells, you would type:

nss.nologin.shell: /sbin/shell_test1 /bin/shell_test2

If this parameter is not configured, the default no-login shells /sbin/nologin and /bin/false are used.

nss.user.conflict.auditlevel

This configuration parameter enables you to override a user's audit level when the user is listed in the user.ignore list and has the use_sysrights audit level. Valid parameter values are:

audit_if_possible
no_audit
audit_required

By default, this parameter is set to audit_if_possible.

nss.user.override.auditlevel

This configuration parameter specifies the default audit level for any users specified for the nss.user.override.userlist without an audit level defined. This parameter replaces the deprecated user.ignore.audit.level parameter.

You can set this parameter to one of the following valid values:

- use_sysrights
- audit_if_possible
- no_audit
- audit_required

The default value is use_sysrights. If there are classic zone users not included in nss.user.override.userlist parameter, the default audit level is the value specified for the nss.alt.zone.auditlevel parameter.

nss.user.override.userlist

This configuration parameter enables you to specify an audit level for a list of users that will bypass Active Directory. In most cases, the auditing service connects to Active Directory to get user profile and audit level information. You can use this parameter to bypass Active Directory, for example, to specify local user accounts that do not have a corresponding user account in Active Directory, but for which you want to audit session activity. This parameter replaces the deprecated user.ignore parameter.

You can specify the parameter value by typing individual entries using the format user_name[:audit_level], separated by spaces, or by using the file: keyword and a file location.

You can set the audit_level to one of the following valid values:

- use_sysrights
- audit_if_possible
- no_audit
- audit_required

The use_sysrights setting indicates that you want to use the audit level information associated with the user's role. If you don't specify an audit level for a user with this parameter, the default audit level is to the audit level you specify for the nss.user.override.auditlevel parameter. For example, you can set the value using individual user name entries like this:

nss.user.override.userlist: maya:use_sysrights tai:no_audit carlos

Alternatively, you can using the file: keyword and a file that has one user_name[:audit_level] per line. For example:

nss.user.override.userlist: file:/etc/centrifyda/user_auditing_classiczones

Be sure to run the dareload command after modifying the configuration file to have the changes take effect.

Note that this parameter is most commonly used to specify the audit level for local user accounts. However, you can use it to specify both local and Active Directory users, if needed. To include Active Directory users in the list of users specified with this parameter, you must specify the Active Directory user's UNIX login name as a parameter value in the user list you define with this parameter.

> **Note**: For computers that have only the Delinea Client for Linux installed, there is a sample file that you can use to specify users outside of Active Directory. The sample file is /etc/centrifyda/nss.user.override.userlist.sample. To point the client to this sample file, include the following line in your centrifyda.conf file:
> nss.user.override.userlist: /etc/centrifyda/nss.user.override.userlist.sample

preferred.audit.store

If your UNIX or Linux computer has multiple IP addresses that match the criteria for multiple audit stores, use the preferred.audit.store parameter to specify the preferred audit store that the DirectAudit agent will use.

If you have this kind of installation and you don't specify the preferred audit store, the collector may or may not connect to the correct audit store.

prefer.site.over.subnet

When the audit and monitoring service is running on a system and it needs to connect to an audit store, you can use this prefer.site.over.subnet parameter to specify the search precedence.

If you set this parameter to true, then the agent looks for an audit store based on the Active Directory sites. If the search fails, then the agent looks for an audit store based on the subnet.

By default, this parameter is false, which means that audit stores are chosen first based on their subnet addresses and then based on the Active Directory sites.

For example:

prefer.site.over.subnet: true

spool.diskspace.logstate.reset.threshold

This configuration parameter specifies a threshold percentage of disk space that is added to the minimum percentage of disk space (set in the spool.diskspace.min parameter) that determines when the information/warning/error log state is reset. Message logging resumes only after the log state is reset.

When disk space drops below the minimum percentage (for example, 10%), a warning is logged. Additional warnings are not logged until disk space has risen above the minimum percentage + threshold percentage (for example, 10% + 2% = 12%), and then drops again below the minimum percentage (10%).

Setting a threshold percentage is useful to prevent unnecessary log messages when disk space hovers near the minimum percentage and would otherwise trigger a log message every time the minimum percentage is crossed.

The default value is 2 percent of disk space.

You can also set this parameter using group policy.

spool.diskspace.min

This configuration parameter specifies the minimum volume of disk space required on the partition containing the offline spool file before spooling stops.

You can set this value as a percentage of the disk space, or you can set it as an exact size. To set the value as an exact size, specify the unit value after the number value. The unit values are not case-sensitive.

For example, to set the minimum volume of disk space to 28 gigabytes, you would enter the following:

spool.diskspace.min: 28GB

You can specify the following unit values:

- B (byte)
- KB (kilobytes)
- MB (megabytes)
- GB (gigabytes)
- TB (terabyrtes)

To specify the value as a percentage, you can either use the percent (%) symbol, or enter a number with no unit value. If you specify a number, make sure that there isn't a space between the number and the unit value; for example, to specify 28 gigabytes enter "28GB" not "28 GB". Including a space, such as "28 GB", is interpreted as a percentage, such as "28%".

The default value is 10 percent of disk space.

You can also set this parameter using group policy.

spool.diskspace.softlimit

This configuration parameter specifies the volume of disk space required on the partition containing the offline spool file to avoid warnings in the log. If available disk falls below the level specified in this parameter, a warning is logged and auditing will continue until disk space falls below the level specified in spool.diskspace.min.

You can set this value as a percentage of the disk space, or you can set it as an exact size. To set the value as an exact size, specify the unit value after the number value. The unit values are not case-sensitive.

For example, to set the minimum volume of disk space to 5 kilobytes, you would enter the following:

spool.diskspace.min: 5 kb

You can specify the following unit values:

- B (byte)
- KB (kilobytes)
- MB (megabytes)
- GB (gigabytes)
- TB (terabyrtes)

To specify the value as a percentage, you can either use the percent (%) symbol, or enter a number with no unit value.

The value of this parameter must be greater than or equal to the value of spool.diskspace.min.

The default value is 12 percent of disk space.

You can also set this parameter using group policy.

spool.maxdbsize

This configuration parameter specifies maximum disk space in bytes to allocate to the offline storage database. Use 0 to designate no limit.

The default value is 0 (unlimited).

You can also set this parameter using group policy.

uid.ignore

This configuration parameter specifies one or more numeric user identifiers (UID) that you want to ignore for authentication and lookup requests in Active Directory. In most cases, you use this parameter to specify local user accounts that do not have a corresponding user account in Active Directory, but for which you want to audit session activity. You can specify the parameter value by typing individual user identifiers, separated by spaces, or by using the file: keyword and a file location. For example, you can set the value using individual UID values like this:

uid.ignore: 0 500 5861

Alternatively, you can using the file: keyword and the sample uid.ignore file that is installed with the Delinea Agent. The sample uid.ignore file ignores the most common default system accounts. For example:

uid.ignore: file:/etc/centrifydc/uid.ignore

If you edit the /etc/centrifydc/uid.ignore file, be sure to run the adreload command after modifying the file to have the changes take effect.

user.ignore

This configuration parameter specifies one or more user names that you want to ignore for authentication and lookup requests in Active Directory. In most cases, you use this parameter to specify local user accounts that do not have a corresponding user account in Active Directory, but for which you want to audit session activity. You can specify the parameter value by typing individual user names, separated by spaces, or by using the file: keyword and a file location. For example, you can set the value using individual user name values like this:

user.ignore: tai carlos games gopher

You can also specify the user's audit level by adding the value after the user names added to user.ignore. For example:

user.ignore: tai carlos:audit_if_possible

Alternatively, you can using the file: keyword and the sample user.ignore file that is installed with the Delinea Agent. The sample user.ignore file ignores the most common default system accounts. For example:

user.ignore: file:/etc/centrifydc/user.ignore

If you edit the /etc/centrifydc/user.ignore file, be sure to run the adreload command after modifying the file to have the changes take effect.

user.ignore.audit.level

This configuration parameter is no longer used except for backward compatibility. It has been replaced by the nss.user.override.auditlevel parameter. This configuration parameter specifies the audit level setting for the user that you want to ignore for authentication and lookup requests in Active Directory. In most cases, you use this parameter when you have specified local user accounts that do not have a corresponding user account in Active Directory, but for which you want to audit session activity. By default, the users you specify for the uid.ignore or user.ignore parameter are audited if auditing is enabled and the auditing service (dad) is running on the local computer (audit if possible). You can disable the auditing of user activity for the users specified by the uid.ignore or user.ignore parameter by setting this parameter value to one (1). For example, if you don't want to audit activity for the users specified in uid.ignore or user.ignore list, set the parameter as follows:

user.ignore.audit.level: 1

You cannot require auditing for the users specified in uid.ignore or user.ignore list because those users would be unable to log on if the auditing service stops running or is removed from a local computer. To prevent users from being locked out, you can only set this parameter to audit if possible (0) or no auditing (1).

Customizing LDAP Proxy Configuration Parameters

This section describes the configuration parameters that affect the use of Delinea direct access LDAP proxy handling on the local host computer.

ldapproxy.cache.credential.expire

This configuration parameter specifies how long ldapproxy holds the ldapproxy credential cache. The cache expires in the specified number of seconds. Default is 300 seconds:

ldapproxy.cache.credential.expires <seconds>

The ldapproxy credential cache retains the user login Active Directory verification.

- If a user logs in within the expiration window, ldapproxy does not require another adclient verification.
- If a user logs in and the ldapproxy credential cache is expired, ldapproxy sends a request for authentication to adclient. adclient can authenticate using its own cache or send to AD for authentication.

Restart centrify-ldapproxy to apply changes.

[root]$ /usr/share/centrifydc/bin/centrify-ldapproxy restart

ldapproxy.netgroup.use.rfc2307nisnetgroup

This configuration parameter specifies which type of netgroup object that the LDAP proxy searches. By default, this parameter is set to false and the LDAP proxy searches the NIS map netgroup objects.

ldapproxy.netgroup.use.rfc2307nisnetgroup false

If you set this ldapproxy.netgroup.use.rfc2307nisnetgroup parameter to true, then the LDAP proxy searches RFC2307 NIS netgroup objects.

Restart centrify-ldapproxy to apply changes.

[root]$ /usr/share/centrifydc/bin/centrify-ldapproxy restart

ldapproxy.performance.log.interval

This configuration parameter specifies the interval between each log dump of ldapproxy events. The interval is number of seconds. The default value, 0, disables the parameter.

ldapproxy.cache.enabled true

ldapproxy.performance.log.interval <seconds>

The ldapproxy statistics collected include:

- Search cache statistics, such as: cache hit, cache negative hit, cache miss, cache expires, cache overflow and object clone error.

  Search cache statistics require the ldapproxy.cache.enabled parameter is set to true and the ldapproxy.performance.log.interval is set to a non-zero value.

- Authentication statistics, such as: authentication requests, satisfied from cache, actual authentication to adclient, and average response time of actual authentication through adclient.

  Authentication statistics require the ldapproxy.performance.log.interval is set to a non-zero value.

Restart centrify-ldapproxy to apply changes.

[root]$ /usr/share/centrifydc/bin/centrify-ldapproxy restart

Important: We are currently migrating this content to this platform. For a complete, working version, please visit our legacy platform here.

The *Group Policy Guide* describes the Delinea group policies that are available in Server Suite for cross-platform access control and privilege management. These group policies allow you to centrally manage computer and user configuration settings through the Microsoft Group Policy Objects.

Intended Audience

This guide is intended for administrators who want to customize the operation of Delinea software by modifying group policies.

This guide is intended as a supplement to the main documentation set and assumes that you have a working knowledge of Delinea architecture and administration and Active Directory group policies.

Using this Guide

Depending on your environment and role as an administrator or user, you may want to read portions of this guide selectively. The guide provides the following information:

- Group policies in Active Directory provides an introduction to group policies, how they are enabled, and how they are applied to Active Directory objects.
- Server Suite group policy overview provides an overview of how Delinea group policies work.
- Adding Delinea settings to Group Policies Objects describes how to add Centrify group policies to a Group Policy Object and how to edit group policy settings.
- DirectControl Settings describes the group policies that control Delinea configuration parameters that are not related to auditing.
- Audit and audit trail settings describes the group policies that control Delinea auditing configuration parameters.
- Additional group policies for UNIX services describes the single-purpose group policies you can add to a Group Policy Object.
- GNOME settings describes the Gnome group policies you can add to a Group Policy Object.
- Mac OS X Settings provides an overview of the group policies available for Mac OS X users and computers.
- Defining custom group policies describes how to create custom administrative templates to implement your own group policies.

Group Policies in Active Directory

This section provides an overview of how to use group policies configuration management in an Active Directory environment. It includes an introduction to the concept of Group Policy Objects on Windows and a summary of how group policies settings are inherited through an Active Directory structure.

The following topics are covered:

Configuring Computer and User Settings

How Group Policies are Applied

Editing a Group Policy Object

Selecting Computer or User Settings

Applying Policies in Nested Organizational Units

Configuring Group Policies to be Refreshed

> **Note**: This section only provides an overview of key concepts for working with group policies and Group Policy Objects. For more complete information about creating and using group policies and working with Group Policy Objects, see your Active Directory documentation. If you are already familiar with group policies and inheritance rules for Group Policy Objects, you can skip this chapter.

Configuring Computer and User Settings

Group policies allow you to specify a variety of configuration options and apply those settings to specific groups of computers and users through Active Directory. In a standard Windows environment, these configuration options control many aspects of computer operation and the user experience, including the user's desktop environment, operations performed during startup and shutdown, local security enforcement, user- and computer-based settings in the local Windows registry, and software installation and maintenance services.

The configuration options available and the settings you make for those options are defined in a **Group Policy Object** (GPO) linked to an Active Directory object. Each Group Policy Object can consist of configuration information that applies to computers, configuration information that applies to users, or sections of policy specifically devoted to each.

Every Group Policy Object includes a default set of **Administrative Templates** and Software and Windows Settings that are created automatically as part of the Group Policy Object. Centrify provides additional templates to manage the Linux, UNIX, and Mac OS X computers. See Adding Centrify Policies from XML Files to learn how to add the Delinea templates to a group policy object.

There are two default Group Policy Objects available when you install or promote a server to be a Windows domain controller:

- Default Domain Controllers Policy
- Default Domain Policy

Your organization may have additional Group Policy Objects customized to suit your environment.

How Group Policies are Applied

Before you can configure any settings by enabling group policies, you must create or select a Group Policy Object where the policies will apply. You can link Group Policy Objects to a specific organizational unit, domain, or site in Active Directory.

To create a new Group Policy Object

1. Open the Group Policy Management console (gpmc.msc).
2. Select a domain, organizational unit, or site, right-click, then select **Create a GPO in this domain, and Link it here**.

   You must have read and write permission to access the system volume of the domain controller and the right to modify the selected site, domain, or organizational unit.
3. Type a name and, optionally, select an existing Group Policy Object to use as a model for the new Group Policy Object, then click **OK**.

Alternatively, you can select a domain, organizational unit, or site in the Group Policy Management console, right-click, then select **Link an Existing GPO** to link an existing Group Policy Object—such as the Default Domain Policy—to the selected domain, organizational unit, or site. Note that you cannot link a Group Policy Object to generic containers—such as the default Users, Computers, or Domain Controllers containers—or to containers you create.

Once you link a Group Policy Object to an organizational unit, domain, or site, the specific policies you enable are applied when computers are rebooted, when users log on, or at the next update interval if you set policies to be periodically refreshed.

Order in which Policies are Applied

You can link Group Policy Objects throughout the hierarchical structure of the Active Directory environment. When you have different policies at different levels, they are applied in the following order unless you explicitly configure them to block inheritance or behave differently:

- Local Group Policy Objects are applied first.
- Site-level Group Policy Objects are applied in priority order.
- Domain-level Group Policy Objects are applied in priority order.
- Organizational Unit-level Group Policy Objects are applied in priority order down the hierarchical structure of your organization, so that the last Group Policy Object used in the one that applies to the Organizational Unit the user or computer resides in.

As this set of rules suggests, a Group Policy Object linked to a site applies to all domains at the site. A Group Policy Object applied to a domain applies directly to all users and computers in the domain and by inheritance to all users and computers in organizational units and containers farther down the Active Directory tree.

A Group Policy Object applied to an organizational unit applies directly to all users and computers in the organizational unit and by inheritance to all users and computers in its child organizational units.

You can modify the specific users and computers the GPO is applied to by choosing a different point in the hierarchy, blocking the default inheritance, using security groups to create Access Control Lists, or defining WMI filters.

> **Note**: There are four group policies (run command, sudo, crontab entries and Linux firewall) that can merge the lines of different group policies to a resulting group policy. For the policies to merge, the policy in each group policy must be enforced. Policies with higher precedence will be placed lower in the resulting multi-line policy. (Ref: CS-21048a)

How the Resulting Policy Set is Determined

The order in which Group Policy Objects apply is significant because, by default, policy applied later overwrites policy applied earlier for each setting where the later applied policy was either Enabled or Disabled. Settings that are Not Configured don't overwrite anything — any Enabled or Disabled setting applied earlier is allowed to persist. You can modify this default behavior by forcing or preventing Group Policy Objects from affecting specific groups of users or computers, but in most cases, you should avoid doing so.

As an example, consider an organization with a single domain called arcade.com which is divided into the following top-level organizational units:

- USA
- Spain
- Korea

Each of these may be divided into lower-level organizational units, indicating major departmental or functional groupings for the top-level organizational unit.

For example, the USA organizational unit may be divided into CorporateHQ, Development, and Sales.

A computer placed in the CorporateHQ organizational unit might then have several different Group Policy Objects applied to it. For example, the arcade.com organization might have a default domain Group Policy Object that applies to all organizational units in the domain, and each organizational unit might also have its own Group Policy Object applied.

The following table illustrates the configuration settings for two computer configuration policies—Windows Update > **Configure Automatic Updates** and Windows Media Player > **Prevent Desktop Shortcut Creation**—for the Group Policy Objects applied to the example organization arcade.com.

Default Domain Policy	arcade.com	Configure Automatic Updates: Enabled with Auto download and notify for install Prevent Desktop Shortcut Creation: Enabled
USA-Specific	USA	Configure Automatic Updates: Not Configured Prevent Desktop Shortcut Creation: Enabled
All Development	CorporateHQ	Configure Automatic Updates: Not Configured Prevent Desktop Shortcut Creation: Disabled

For example, if you were managing the default domain policies used in this example, you would:

1. Start Active Directory Users and Computers.
2. Right-click the domain, arcade.com, then click **Properties**.
3. Click the **Group Policy** tab.
4. Select the **Default Domain Policy**, then click **Edit** to open the Default Domain Policy in the Group Policy Object Editor.
5. Click **Computer Configuration > Administrative Templates > Windows Components > Windows Update > Configure Automatic Updates** to **Enabled** and set the **Auto download and notify for install** update option and click **OK**.
6. Click **Computer Configuration > Administrative Templates > Windows Components > Windows Media Player > Prevent Desktop Shortcut Creation** to **Enabled** and click **OK**.

When all of the policies described in the table are applied in their default order, a computer in the CorporateHQ organizational unit would be configured with the following policy settings:

- Configure Automatic Updates: **Enabled** with **Notify for download and notify for install**
- Prevent Desktop Shortcut Creation: **Disabled**

The User Configuration policies applied in a Group Policy Object are also determined by the organizational unit in which a UNIX user is a member. For example, if you define separate User Configuration policies in a Group Policy Object linked to the USA organizational unit, you must also add the users to this organization unit for the policies to apply. For more information, see Applying Policies in Nested Organizational Units.

Editing a Group Policy Object

Any time you create a new Group Policy Object for an organizational unit, domain, or site, it includes a set of default configuration options for computers and users. Initially, all of these default configuration options are defined as “Not configured” or “Not defined” and have no effect. You can then enable the specific policies you want to use for the organizational unit, domain, or site linked to the current Group Policy Object by opening the Group Policy Object in the Group Policy Management Editor.

To edit a specific Group Policy Object

1. Open Administrative Tools, Group Policy Management (gpmc.msc).
2. Expand the Forest and Domains nodes to select a domain,
3. Expand Group Policy Objects for the domain.
4. Select an existing Group Policy Object—such as Default Domain Policy—then right-click and select **Edit**.

The default templates in Group Policy Objects do not include Centrify policies for Centrify-managed computers. For information about adding Centrify policies to Group Policy Objects, see Adding Centrify Policies from XML Files.

Selecting Computer or User Settings

Group Policy Objects consist of two types of group policy settings:

- **Computer Configuration** policies define the startup and shut down operations and other computer-specific behavior. These configuration settings apply to the computers regardless of the user account that logs on to the computer.
- **User Configuration** policies define log-on and log-off operations and other user-specific behavior. These configuration settings apply to the user account regardless of the computer the user logs on to. With these settings, users can move from computer to computer with a consistent profile.

Because the computer and user group policies contain different configuration settings, they don't affect each other directly. In planning how to implement group policies, however, you need to keep in mind which policies must be computer-based and which must be user-based. In many cases, the same group policy might be available as both a computer configuration policy and a user configuration policy. In those cases, you need to decide whether the policy is best applied to computers and all users who log on or to individual users when logging on, regardless of the computers they use.

Applying Policies in Nested Organizational Units

In many production environments, user accounts are most often defined in a parent organizational unit and computers are often placed in a child organizational unit (OU). If you have a Group Policy Object that is linked to the child organizational unit for computer policies, but the user accounts are in a parent organizational unit, the user configuration policies linked to the child organizational unit are not applied to the users when they log in to the computers in the child organizational unit. Instead, the user configuration policies linked to the child OU only apply to the users who are in that child OU.

There are two ways to apply different user configuration policies at lower levels in the organizational unit tree:

- Set the User Configuration policies at the parent level and then configure the child organizational unit to inherit the group policies from the parent.
- Enable the **User Group Policy loopback processing mode** group policy in the Group Policy Object linked to the child organizational unit to implement different user configuration policies at each level.

The **User Group Policy loopback processing mode** group policy is located under Computer Configuration, Policies, Administrative Templates, System, Group Policy. When it is enabled, Active Directory applies the Group Policy Object settings defined for the computers in the child organizational unit to all users.

Enable the Loopback Policy

1. Open Administrative Tools, Group Policy Management (gpmc.msc).
2. Select the Group Policy Object linked to the child organizational unit, right-click, then select Edit.
3. Expand Computer Configuration to view policies under Group Policy.
4. Double-click **User Group Policy loopback processing mode** group policy, the select **Enabled**.

   For Mode, select **Replace** if you defined a whole new set of policies or **Merge** if you are just modifying a subset of policies.

Delinea

Configuring Group Policies to be Refreshed

The computer portion of a Group Policy Object is normally applied any time you restart a computer that receives group policies. The user portion of a Group Policy Object is normally applied any time a user logs on to a computer. Both the computer and user portions of a Group Policy Object can also be configured to refresh automatically at a set interval.

To configure the refresh interval and the conditions for refreshing group policies, use the policies listed under **Computer Configuration > Administrative Templates > System > Group Policy** and **User Configuration > Administrative Templates > System > Group Policy** of a Group Policy Object.

If you configure your Group Policy Objects to refresh periodically, at the interval you specify, the computer contacts Active Directory to get the Group Policy Objects that apply and configures itself with the appropriate settings. If policies are refreshed at a set interval, users can change their configuration settings or their computers' configuration settings, but the changes will be overridden when the group policies are refreshed at the next interval.

If you configure the refresh policy settings for users or computers, the refresh policy applies to both Windows and agent-managed computers and users.

Server Suite Group Policy Overview

This section describes how Server Suite maps the policy settings defined in a Group Policy Object to configuration settings for Server Suite-managed computers and users.

The following topics are covered:

Mapping Settings to a Virtual Registry

In the Windows environment, most of the configuration settings defined in a Group Policy Object are implemented through entries in the local Windows registry. For Linux, UNIX, and Mac OS X computers and users, however, local configuration details are typically defined using a set of configuration files stored in the /etc directory. In addition, the Window and Linux, UNIX, and Mac OS X environments have different configuration requirements, and consequently require different settings to be available through group policy.

To address these differences, Centrify provides its own group policies that allow administrators to use Group Policy Objects to configure settings for Centrify-managed computers and users. To enable you to use Group Policy Objects to configure settings for Linux-, UNIX-, and Mac OS X-based computers and users, Centrify...

- Provides its own **administrative templates** (.xml and .admx files) that define Linux-, UNIX-, and Mac OS X-specific configuration settings.
- Uses the adclient daemon to collect configuration details from Active Directory based on the Group Policy Objects applied for the current computer or user and create a **virtual registry** of those configuration settings on the local Linux, UNIX, or Mac OS X computer.
- Runs local programs that map the configuration details in the virtual registry to the appropriate configuration file changes on the local Linux, UNIX, or Mac OS X computer.

The virtual registry is a collection of files that contain **all** of the group policy configuration settings from the group policies applied to the computer through the group policy hierarchy, including settings that apply only to Windows computers. Because the files that make up this virtual registry are not native to the Linux, UNIX, or Mac OS X environment, the Centrify software then uses a set of **mapping programs** to read the files, determine the settings that are applicable to Linux, UNIX, or Mac OS X computers and users, and make the appropriate changes in the corresponding Linux, UNIX, or Mac OS X configuration files to implement the configuration specified. The mapping programs ignore any Windows-specific settings that have been applied and only map the settings that are appropriate for the Linux, UNIX, or Mac OS X environment.

> **Note**: The virtual registry only supports the group policies that are implemented through registry settings. Group policies that are implemented in other ways, for example, by running an executable script on each computer, aren't supported.

The authentication service daemon, adclient, retrieves policy settings from the Active Directory domain controller and starts the program runmappers (/usr/share/centrifydc/mappers/runmappers). The runmappers program runs the individual mapping programs that are stored in the /usr/share/centrifydc/mappers/machine and /usr/share/centrifydc/mappers/user directories. Those individual mapping programs read settings from the virtual registry and write them as the appropriate settings in application-specific configuration files.

The individual mapping programs also keep track of local changes that conflict with group policy settings, so those changes can be restored if the computer is removed from the domain, or if the configuration setting is removed from a Group Policy Object.

Configuring Settings in Administrative Templates

Administrative templates are stored as files with the.xml or .admx extension in the system volume and are used to define a specific set of configuration options. For most of the configuration settings that apply to Linux, UNIX, or Mac OS X users or computers, you must use Centrify group policy administrative templates. To apply a group policy setting, you must add the template that defines the group policy to a Group Policy Object; see Adding Centrify Policies from XML Files.

In addition, every Group Policy Object includes a default set of Administrative Templates. The default administrative templates provide configuration options for Windows users and computers. In a few cases, however, settings you can configure in the default administrative templates do apply to Centrify-managed computers and users. For information about Windows settings that can be applied to Linux, UNIX, and Mac OS X users and computers, see Using Standard Windows Group Policies.

Mapping Computer Configuration Policies

The Centrify Agent, adclient, determines the group policies that apply to Centrifymanaged computers using the same rules for inheritance and hierarchy that apply to Windows computers. When the Linux, UNIX, or Mac OS X computer starts or when the computer policies are refreshed, adclient:

- Contacts Active Directory.
- Checks for the Group Policy Objects that are linked to each organizational unit of which the local computer is a member.
- Determines all of the configuration settings that apply to the local computer, and retrieves those settings from the System Volume (SYSVOL).
- Writes all of the configuration settings to a virtual registry on the local computer.
- Starts the runmapperS program to initiate the mapping of configuration settings using individual mapping programs for computer policies.

The mapping programs in the /usr/share/centrifydc/mappers/machine directory then read the virtual registry for the appropriate Linux-, UNIX-, or Mac OS X-specific computer configuration settings and locate the appropriate configuration files to change, then modify those files accordingly.

After the computer starts, the adclient daemon will periodically check with Active Directory to determine the current group policy settings for the computer unless you disable group policy updates.

Mapping User Configuration Policies

The adclient daemon determines the group policies that apply to Linux, UNIX, or Mac OS X users using the same rules for inheritance and hierarchy that apply to Windows users. When a user logs into an agent-managed computer, the adclient process detects the log-in and does the following:

- Contacts Active Directory.
- Checks for the Group Policy Objects that are linked to each organizational unit the user is a member of.
- Determines all of the configuration settings that apply to the user account, and retrieves those settings from the System Volume (SYSVOL).
- Writes all of the configuration settings to a virtual registry on the local computer.
- Starts the runmappers program to initiate the mapping of configuration settings using individual mapping programs for user policies.

The mapping programs in the /usr/share/centrifydc/mappers/user directory then read the virtual registry for the appropriate Linux-, UNIX-, or Mac OS X-specific user configuration settings and locate the appropriate configuration files to change, then modify those files accordingly.

After the user has logged on, the adclient daemon will periodically check with Active Directory to determine the current group policy settings for the user unless you disable group policy updates.

Editing Configuration Settings Manually

Many of the group policies are used to modify the parameter values in the authentication service configuration file /etc/centrifydc/centrifydc.conf. When you make changes to a group policy setting, the change is reflected in the /etc/centrifydc/centrifydc.conf file on each joined Linux, UNIX, or Mac OS X computer after the following events:

- The computer restarts.
- The computer configuration policies refresh at the next update interval.
- You run the adgpupdate command.

If you enable Centrify group policies, you do not need to manually edit the configuration parameters in the /etc/centrifydc/centrifydc.conf file. In some rare cases, however, you may find it useful to customize these parameters on a particular computer. For example, you can use configuration parameters to temporarily disable group policies for users, computers, or both, on a computer.

For more information about customizing behavior using the Centrify configuration files and configuration parameters instead of group policies, see the *Configuration and Tuning Reference Guide*.

Updating Configuration Policies Manually

Although there are Windows group policy settings that control whether group policies should be refreshed in the background at a set interval, Delinea also provides a UNIX command line program, adgpupdate, to manually refresh group policy settings at any time. With this command, you can specify whether you want to refresh computer configuration policies, user configuration policies, or both.

When you run adgpupdate, the adclient process does the following:

- Contacts Active Directory for computer configuration policies, user configuration policies, or both. By default, adclient collects both computer and user configuration policies.
- Determines all of the configuration settings that apply to the computer, the current user, or both, and retrieves those settings from the System Volume (SYSVOL).
- Writes all of the configuration settings to a virtual registry on the local computer.
- Starts the runmappers program to initiate the mapping of configuration settings using individual mapping programs for user and computer policies.
- Resets the clock for the next refresh interval.

For more information about using the adgpupdate command, see the adgpupdate man page.

Using Standard Windows Group Policies

Every Group Policy Object includes default administrative templates for user and computer configuration. Most of the settings in the default administrative templates only apply to Windows computers and Windows user accounts. However, there are a few of these common Windows configuration settings that can be applied to Centrify-managed computers and users. These configuration options are not duplicated in Centrify administrative templates.

You can set the following standard Windows group policy options for Centrify-managed computers and users:

Computer Configuration > Policies > Administrative Templates > System > Group Policy	* Turn off background refresh of Group Policy * Group Policy refresh interval for computers
Computer Configuration > Policies > Administrative Templates > System > Windows Time Service > Time Providers	* Global Configuration Settings - MaxPollInterval
Computer Configuration > Policies > Administrative Templates > System > Windows Time Service > Time Providers	* Enable Windows NTP Client This policy specifies that adclient poll the domain NTP server to synchronize the clock of the local computer. This policy modifies the adclient.sntp.enabled parameter in the centrifydc.conf configuration file. If you disable this policy, adclient does not attempt to synchronize the computer with the domain NTP server. The computer uses the local NTP policies, as defined in ntp.conf. Whether you enable the policy or not, no settings are changed in the ntp.conf file.
Computer Configuration > Policies > Administrative Templates > Windows Components > Smart Card > Allow certificates with no extended key usage certificate attribute	* Allow sctool to obtain Kerberos credentials even though the certificate does not have the extended key usage attribute.
Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options	Interactive logon: Message text for users attempting to log on Interactive logon: Prompt user to change password before expiration
Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Password Policy	Enforce password history Maximum password age Minimum password age Minimum password length Password must meet complexity requirements Store passwords using reversible encryption
Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Trusted Root Certification Authorities	Specifies the trusted root CA certificate to use
User Configuration > Policies > Administrative Templates > System > Group Policy	Group Policy refresh interval for users

Reporting group policy settings

On Windows computers, you can use the optional Group Policy Management Console to see the results of group policy settings for a specific computer or user, including Server Suite-managed computers and users.

You can also review the results of group policy settings for a Server Suite-managed computer or a specific user by viewing the gp.report file locally on the computer. This report is automatically updated at each group policy update interval. By default, the gp.report for computer configuration is located in the /var/centrifydc/reg/machine directory and the gp.report for user configuration is located in the /var/centrifydc/reg/users/_username_ directory.

Generating a report of Delinea group policies

Delinea includes a command-line utility on Linux and UNIX systems called adgpresult that you can use to generate a report of all the group policy settings that are in effect for the local computer, the current user, or a specified user.

If you have applied a GPO (group policy object) to a site, domain, or organizational unit that includes a Delinea-managed computer, then you can use the adgpresult command to see all the computer and user configuration policies that have been applied.

The command displays a Resultant Set of Policies similar to the Microsoft Windows gpresult program.

Example usage and options

```
adgpresult [--all] [--machine] [--user __user_name__]
```

You can use the following options with this command:

-a, --all I Option displays both the computer and user group policy settings that are in effect for the local computer and the current user account.

-m, --machine I Option displays only the computer group policy settings that are currently in effect on the local computer.

-u, --user I The --user option displays only the user group policy settings that are in effect for the currently logged on user or for the user specified by the user_name argument.

Example:

To report only the computer configuration policies and save the results to a file, you could type a command similar this:

```
adgpresult --machine \> /tmp/unix-rsop-rhel6
```

Adding Centrify Settings to Group Policy Objects

This chapter describes how to add Centrify-specific group policies to a Group Policy Object and how to set policies for Centrify-managed computers and users.

The following topics are covered:

Configuring Audit Event Logging Location by Group Policy

Audit trail group policies are located in category-specific subfolders (such as **Audit Analyzer Settings**, Audit Manager Settings, and so on). Additionally, a **Delinea Global Settings** subfolder contains group policies that you can set at a global level.

Any category-specific audit trail targets that you set (for example, **Audit Manager Settings** › **Send audit trail to log file**) override global audit trail targets (for example, Delinea Global Settings › Send audit trail to log file). Each subfolder in **Delinea Audit Trail Settings** contains the same set of group policies.

> **Note**: To send audit trail events to both the database and the local logging facility, enable both of these group policies.

Send Audit Trail to Audit Database

Enable this group policy to specify that audit events for this component **Audit Analyzer, Audit Manager**, and so on—are sent to the active audit store database.

See the **Explain** tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Send Audit Trail to Log File Enable this group policy to specify that audit events for this component— such as **Audit Analyzer, Audit Manager**, and so on—are sent to the local logging facility (syslog on UNIX systems, Windows event log on Windows systems).

See the Explain tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Set Global Audit Trail Targets

Specify the target for audit trail information.

If you set this group policy to **Not configured** or **Disabled**, the destination of audit trail information depends on which version of DirectAudit is installed. If DirectAudit 3.2 or later is installed, audit trail information is sent to the local logging facility and DirectAudit. If a DirectAudit version earlier than 3.2 is installed, audit trail information is only sent to the local logging facility.

If you set this group policy to **Enabled**, you can specify the target for audit trail information. Possible settings are:

- 0 (Audit information is not sent.)
- 1 (Audit information is sent to Delinea Audit & Monitoring Service. This capability is supported by DirectAudit version 3.2 and later.)
- 2 (Audit information is sent to the local logging facility, either syslog on UNIX systems or Windows event log on Windows systems.)
- 3 (Audit information is sent to both DirectAudit and the local logging facility.)
- This group policy modifies the audittrail.targets setting in the agent configuration file.

Adding Administrative Templates to a Group Policy Object

A Group Policy Object (GPO) consists of configuration information that applies to computers, configuration information that applies to users, or sections of policy specifically devoted to each. You can extend the configuration options provided by any Group Policy Object by adding Centrify-provided or custom administrative templates to the object. For example, you can add configuration settings for Centrifyagents to a Group Policy Object by adding the centrifydc_settings.xml administrative template. Other administrative templates can be added to control other settings, such as Mac OS X system preferences, if they apply to your environment.

Installing Centrify Group Policy Templates

When you install Access Manager using the installation wizard and you specify that all components be installed, the Centrify group policy templates are included in the installation. See "Install Access Manager and update Active Directory" in the *Administrator's Guide for Windows* for details about using the Access Manager installation wizard.

> **Note**: For details about where the Centrify group policy templates reside after they are installed, see Adding Centrify Policies from XML Files.

Because Centrify group policy templates and extensions are packaged separately from other Access Manager components, you have the following options if you prefer to install group policy templates and extensions separately from Access Manager:

- You can install Centrify group policy templates and extensions on any Windows domain computer without also installing Access Manager on the computer.
- You can install Access Manager on any Windows domain computer without also installing Centrify group policy templates and extensions on the computer.

The group policy template and extension package has its own .exe and .msi installer files, so that you can install group policy templates and extensions interactively through an installation wizard (by executing the .exe file) or silently from the command line (by executing the .msi file). Additionally, you can select or de-select the group policy template and extension component for installation when you run the Access Manager installation wizard.

For details about installing group policy templates and extensions separately from Access Manager, see "Install group policy extensions separately from Access Manager" in the *Administrator's Guide for Windows*.

Template File Formats

Centrify provides templates in both XML and ADMX format. In most cases, it is best to use the XML templates, which provide greater flexibility, such as the ability to edit settings after setting them initially, and in many cases contain validation scripts for the policies implemented in the template.

However, in certain cases, you may want to add templates by using the ADMX files. For example, if you have implemented a set of custom tools for the Windows ADMX-based policies, and want to extend those tools to work with the Centrify policies, you can implement the Centrify policies by adding the ADMX template files. You should note, however, that ADMX templates do not support extended ASCII code for locales that require double-byte characters. For these locales, you should use the XML templates.

Selecting a Group Policy Object for Centrify Settings

Depending on the requirements of your organization and how you have linked existing Group Policy Objects to sites, domains, and organizational units in your Active Directory forest, you might want to use one of the default Group Policy Objects, use a Group Policy Object you have created specifically for your organization, or create a new Group Policy Object that is specifically for Centrify settings.

If you have created an organizational structure for Centrify objects as described in the *Planning and Deployment Guide*, creating a new Group Policy Object specifically for Centrify policies gives you the most flexibility and control over the configuration settings for managed computers and the operation of Centrify software. In deciding whether to create a new Group Policy Object or use an existing Group Policy Object, you should consider where policies should be applied. You can link Group Policy Objects to sites, domains, or organizational units to control the scope of the policies you set.

If you prefer to minimize the number of Group Policy Objects you deploy, you can add Centrify settings to one of default Group Policy Objects that are installed on the Windows domain controller:

- Default Domain Controllers Policy
- Default Domain Policy

You can add Centrify settings to any Group Policy Object regardless of whether you have any settings configured or applied to Windows users and computers. Settings that apply to Centrify-managed computers only affect computers where the Centrify Agent is installed.

Linking a Group Policy Object to an Organizational Unit

You can link a Group Policy Object to an organizational unit, domain, or site using the Group Policy Management Console. To set group policies for a selected Active Directory site, domain, or organizational unit, you must have read and write permission to access the system volume of the domain controller and the right to modify the selected directory object.

If you have created an organizational structure for Centrify as described in the *Planning and Deployment Guide*, the most natural place to link a Group Policy Object is the top-level container of that organizational unit structure, for example, the Centrify container.

Create and Link a Group Policy Object for Centrify Settings

1. Click **Start > Administrative Tools > Group Policy Management**.
2. Select the Centrify organizational unit, right-click, then select **Create a GPO in this domain, and Link it here**.
3. Type a name for the new Group Policy Object, for example, Centrify Policy, then click **OK**.

If you want to apply group policies to lower levels in the organizational structure, you can do so by linking Group Policy Objects to lower level organizational units. For example, if you created a separate organizational unit for zone computers, you can link a Group Policy Object to that organizational unit. However, you cannot link Group Policy Objects to containers (CN).

Using Security Filtering for Group Policies

You can use Active Directory security groups and group policy security filtering if you want to restrict the policies applied to subsets of zone computers or users. By creating an Active Directory security group and setting security filtering for a Group Policy Object, you can achieve fine-grain control over where group policies are applied within the Centrify organizational unit structure. For example, you can create an Active Directory group called europe that has a specific set of computers in it, then restrict the application of group policies to that group.

To enable security filtering of group policies:

1. Create the Active Directory security group with the appropriate members.
2. Open the Group Policy Management Console and select the Group Policy Object for which you want to enable filtering.
3. On the Scope tab, under Security Filtering, click **Add**.
4. Be certain that 'Group' appears in **Select this object type**; if not, Click **Object Types** and select **Groups**.
5. Type all or part of the name for the group you created for filtering, click **Check Names**.

   If more than one group is returned, select the appropriate group, then click **OK**.
6. Click **OK** to link the security group to scope of the Group Policy Object.

Adding Policies from XML Files

In most cases, you should add Centrify policies from XML templates to the Group Policy Object you are using for Centrify settings. The XML-based format is the current standard for group policy templates.

To add Centrify group policies from Centrify XML templates:

1. Click **Start > Administrative Tools > Group Policy Management**.
2. Expand the appropriate site, domain, or organizational unit to select Group Policy Object you want to use for Centrify policies, right-click, then click **Edit**.

   For example, expand the top-level Centrify organizational unit to select the Centrify Policy object, right-click, then click **Edit**.
3. In the Group Policy Management Editor, expand Computer Configuration and Policies.
4. Select Centrify Settings, right-click, then click **Add/Remove Templates**.
5. In the Add/Remove Templates dialog box, click **Add**.
6. In most cases, the directory with the templates is already selected and the following Centrify templates are listed:

   - centrify_gnome_settings
   - centrify_linux_settings
   - centrify_mac_settings
   - centrify_unix_settings
   - centrifydc_fips
   - centrifydc_settings
   - centrifyds_settings

   If the templates are not listed, navigate to the group policy directory under the Access Manager installation directory. For example, if you installed files in the default location, navigate to the following directory:

   C:\Program Files\Common Files\Centrify Shared\Group Policy Management Editor Extension\policy

   If you want to add templates for auditing, navigate to the Centrify Audit & Monitoring Service installation directory. For example, if you installed files in the default location, navigate to the following directory:

   C:\Program Files\Centrify\Audit\AuditManager
7. Select the Centrify templates you want to use, then click **Open**.
8. In the Add/Remove Templates dialog box, click **OK** to add the new templates.

   Group policies for access control and privilege management are listed under **CentrifySettings**. You can expand this node and the categories below it to explore the group policies available.

   Group policies for auditing are listed under **Centrify Audit Settings**. You can expand this node and the categories below it to explore the group policies available.

   By default, all group policies are set to "Not configured."

Adding Templates after an Upgrade

To make any new policies available after you upgrade Centrify software, you must add new versions of the templates you use after you upgrade the Access Manager or the auditing console. To add new versions of the templates after an upgrade, repeat Linking a Group Policy Object to an Organizational Unit to Adding Centrify Policies from XML Files. If you see the message, The selected XML file already exists. Do you want to overwrite it?, click **Yes** to overwrite the old template file with the new template and make any new or modified group policies available. Overwriting the template does not affect any configuration settings that have been applied. Policies that you have enabled remain enabled.

Enabling Delinea Policies

By default, all group policies, including Delinea group policies are set to Not configured. You can selectively enable the specific computer and user policies you want to use. Most of the Delinea group policies set configuration parameters on managed computers. If you choose to enable any of these group policies, you should be familiar with the corresponding configuration parameters described in the *Configuration and Tuning Reference Guide*.

To enable and configure Delinea settings:

1. Open the Group Policy Management console.
2. Select the Group Policy Object to which you have added Delinea policies, right-click, then select **Edit**.
3. Expand **Computer Configuration > Policies > Centrify Settings**.
4. Select a policy name, right-click, then select **Properties**.
5. Click **Enabled**.

   Depending on the policy, you might need to select values or provide other information to complete the configuration. For more information about the policy and how to set configuration options, click the **Explain** tab. For information about limitations to the values that you specify, see Centrify Policy Limitations later in this chapter.
6. Click **Apply** after making the change.

The policies you enable are applied when computers in the site, domain, or organizational units are rebooted, users next log on, or at the next update interval.

Delinea Policy Limitations

Some Delinea group policies allow you to select values from a list when you enable the group policy. Depending on how the list is configured, in some cases you cannot select more than 999 items from the list.

For example, if you enable the **Specify AD users allowed in Auto Zone** group policy, you are prompted to specify the names of AD users. You can specify AD user names by typing them, by specifying a file containing a predefined list of user names, or by selecting them from the list of all AD users. Because of the way in which the AD user list is configured, you cannot select more than 999 users from it. If you attempt to select more than 999 users, the following message is displayed:

```
Please enter 0 to 999 entries for User
```

This limitation applies to several other group policies in addition to **Specify AD users allowed in Auto Zone**.

It is generally not advisable to select 1000 or more items from a list to define a data set in a group policy. Instead, whenever possible you should use groups or a file containing a predefined list of items to define a large data set.

DirectControl Settings

The following table summarizes the group policies listed directly under **Centrify Settings › DirectControl Settings**. The full descriptions follow the table.

Add centrifydc.conf Properties	Add configuration parameters to centrifydc.conf configuration file.
Maintain DirectControl 2.x Compatibility	Maintain access for legacy users or computers.
Merge Local Group Membership	Merge local group membership from /etc/group into the zone group membership for groups that have the same name and GID.
Prefer Authentication Credentials Source	Instruct adclient to authenticate the user using the cached credentials.
Set LDAP Fetch Count	Specify the number of objects to obtain in a single LDAP request.
Set Password Cache	Control the caching of user passwords.
Set User Mapping	Map a local user account to an Active Directory account.
Use FIPS 140-2 Compliance Algorithms	Select the algorithms used for the authentication protocols.

Additional group policies for DirectControl Settings are organized under the following subnodes:

- Account Prevalidation - Contains policies to manage prevalidation of users and groups for disconnected systems.
- Adclient Settings - Contains policies to control certain aspects of the operation of the agent on managed computers.
- Auto Zone Group Policies - Contains policies to control certain aspects of the operation of the agent on machines that are joined to Auto Zone.
- Dzdo Settings - Contains policies to control certain aspects of the operation of dzdo and sudo.
- Group Policy Settings - Contains policies to manage the execution of the Centrify group policy mapping programs.
- Kerberos Settings - Contains policies to manage the Kerberos configuration. You can use these settings to control updates to the Kerberos configuration files and credential renewal.
- Local Account Management Settings - Contains policies to control agent management of local users and groups.
- Logging Settings - Contains policies to control logging policy settings. You can use these settings to specify the syslog facility to use for logging different adclient processes and to control the amount of memory to use to queue log messages.
- Login Settings - Contains policies to control login and local account access. You can use these settings to grant or deny access to specific users and groups or to ignore Active Directory authentication for some users and groups.
- MFA Settings - Contains policies for configuring multi-factor authentication in classic zones and Auto Zones. You can use these settings to specify which users or groups require a two-step authentication procedure for login, define rescue users that can log in when multi-factor authentication is unavailable, and to specify a cloud URL to be used in multi-factor authentication.
- Network and Cache Settings - Contains policies to specify the maximum period for client connection time-outs and object expiration intervals. You can use these settings to determine how long to wait for a response when connecting to Active Directory and how long objects should be kept in the local cache.
- NIS Daemon Settings - Contains policies to control operation of the Centrify Network Information Service (adnisd) on the local host computer. The adnisd service provides a mechanism for the Centrify Agent to respond to NIS client requests from other computers not managed by Centrify software.
- NSS Overrides - Contains policies to specify the passwd or group override entries you want to use in place of the entries in the local /etc/passwd or /etc/group files. You can use these settings to provide fine-grain control of the users and groups who can use the computer and to override the user ID, group ID, default shell, or home directory for specific login accounts or groups.
- PAM Settings - Contains policies to customize the behavior of the Centrify PAM module.
- Password Prompts - Contains policies to customize the prompts displayed when Active Directory users are prompted to provide their password. You can use these settings to change the text displayed when Active Directory users log in or change their password.
- Sudo Settings - Contains policies to control certain aspects of the operation of sudo.
- User's Initial Group ID - Contains policies to control group numbers. You can use this setting to specify the default group identifier for new users.

Add centrifydc.conf properties

Use the **Add centrifydc.conf properties** group policy to add configuration parameters to the agent configuration file. Although you can set many configuration parameters and values by using the associated group policy, not all configuration parameters have an associated group policy. The **Add centrifydc.conf properties** group policy enables you to specify any configuration parameter and its value.

See the *Configuration and Tuning Reference Guide* for a list of all configuration parameters.

To use this group policy, select **Enabled**, then click **Add**. Enter a property name and property value. For example, to change the adnisd update interval to 10 minutes:

Property name: nisd.update.rate

Property value: 600

Be careful when adding parameters because there is no error checking. If you enter a nonexistent property name or invalid value, the parameter and value will be added to the configuration file as-is. An invalid parameter name will simply be ignored but an invalid value could cause configuration problems.

Enable Active Directory PAM Privilege Escalation Feature

Use the **Enable Active Directory PAM Privilege Escalation Feature** group policy to specify if the Microsoft Privileged Access Management (PAM) Privilege Escalation feature is supported or not within the Delinea environment.

If this policy is Enabled, then, when an Active Directory user logs in, the configured privilege that's granted to the user through PAMGroup takes effect until the granted period has elapsed.

The Microsoft PAM Privilege Escalation feature specifies if Delinea DirectControl uses Microsoft PAM Privilege Escalation feature in the computer.

This group policy modifies the `microsoft.pam.privilege.escalation.enabled` setting in the agent configuration file.

Maintain DirectControl 2.x compatibility

Use the **Maintain DirectControl 2.x compatibility** group policy if you have legacy users or computers who were given access using the console, version 2.x.x.

If all of your Active Directory users are enabled for Linux, UNIX, or Mac OS X access using the console, version 3.0 or later, you should leave this policy as not configured.

This group policy modifies the `adclient.version2.compatible` setting in the agent configuration file.

Merge Local Group Membership

Use the **Merge local group membership** policy to determine whether to merge local group membership from the /etc/group file into the zone group membership for groups that have the same name and GID. For example, if the agent retrieves the membership list of kwan, emily, and sam for the group profile with the group name performx1 and GID 92531 from Active Directory and there is also a local group named performx1 with the GID 92531 with users wilson and jae, the merged group would include all five members (kwan, emily, sam, wilson, jae).

This group policy modifies the adclient.local.group.merge setting in the agent configuration file. By default, the parameter associated with this policy is set to false to prevent unexpected results.

Be careful when enabling this policy, because it violates normal NSS behavior and, therefore, may have unexpected side effects. You should analyze your environment carefully, and determine that you can safely merge local and Active Directory group profiles before enabling this policy.

Prefer Authentication Credentials Source

Use the **Prefer Authentication against cached credentials** policy to authenticate the user using the cached credentials first, regardless of the current connectivity state with the Active Directory domain controller.

By default, the parameter associated with this policy is set to false. You can enable this policy to reduce traffic on slow networks. However, if the Active Directory credentials are not synchronized with the cached credentials, you run the risk of undesired side affects when the computer is online.

This group policy modifies settings in the agent configuration file. For more information about the configuration file and this configuration settings, see adclient.prefer.cache.validation.

Set LDAP Fetch Count

Use the **Set LDAP fetch count** group policy to specify the number of objects to obtain in a single LDAP request. You can use this group policy to optimize the number of objects to suit your environment.

If you select **Enabled** for this group policy, you can then set the number of objects to obtain in a single LDAP request by balancing speed and memory usage against network bandwidth and latency. As you increase the number of objects included in an LDAP request, you may improve the overall performance by decreasing the number of connections to Active Directory and reducing the overall demand on the server, but you increase the RAM used by the agent. If you decrease the number of objects included in an LDAP request, you may reduce overall performance because of the additional network traffic, but decrease the memory used by the agent.

On faster networks, you can safely retrieve a small number of objects. On slower networks or when retrieving information for large groups (for example, groups with more than 1000 users), you may want to increase the value for this parameter.

This group policy modifies the adclient.fetch.object.count setting in the agent configuration file.

Set Password Cache

Use the **Set password cache** group policy to control the handling of user passwords. By default, the Centrify Agent stores a UNIX-style MD5 hash of each user's password in the cache when the user is authenticated during login. Storing the password hash allows previously authenticated users to log on when the computer is disconnected from the network or Active Directory is unavailable.

If you select **Enabled** for this group policy, you can set the following options:

- **Allow Password storage** Allow specified users to have their password hash stored in the cache. If you set this option and specify a list of users, only those users can log on when the computer is disconnected from the network or Active Directory is unavailable. To list the specific users allowed to have their password hash stored, type the user names separated by commas or spaces, or click **List**, then **Add** to browse and select Active Directory users to add.

  This option modifies the adclient.hash.allow parameter in the agent configuration file. By default, all users have their password hash stored.

- **Deny Password storage** Prevent specified users from having their password hash stored. If you set this option and specify a list of users, only those users are prevented from logging on when the computer is disconnected from the network or Active Directory is unavailable. To list the specific users who should not have their password hash stored, type the user names separated by commas or spaces, or click **List**, then **Add** to browse and select Active Directory users to add. This setting overrides "Allow Password storage".

  This option modifies the adclient.hash.deny parameter in the centrifydc.conf agent configuration file. By default, all users have their password hash stored.

- **Cache life** Specify the number of days a password hash for any user can be stored in the cache before it expires. A value of zero (0) specifies that the password hash should never expire. When you enable this policy, a value of 7 (days) appears in the field. You can accept this value or enter a different value up to 9999.

  This option setting modifies the adclient.hash.expires parameter in the centrifydc.conf agent configuration file. The default setting for this parameter is 0, which means that by default, the cache does not expire.

For more information about the configuration file and these configuration settings, see adclient.hash.allow, adclient.hash.deny, and adclient.hash.expires in the *Configuration and Tuning Reference Guide*.

Set User Mapping

Use the **Set user mapping** group policy to map a local Linux, UNIX, or Mac OS X user account to an Active Directory account. Local user mapping allows you to set password policies in Active Directory even when a local Linux, UNIX, or Mac OS X account is used to log in. This group policy is most commonly used to map local system or application user accounts on a computer to a different Active Directory account and password, so that you can enforce password complexity rules for the account, but it can be used for any local user account.

When you select **Enabled** for the Set user mapping group policy, you can then click **Show** to add or remove user accounts.

To add mapped user accounts to the policy, click **Add**. You can then type the Linux, UNIX, or Mac OS X user account name in the first field and the Active Directory account name to which you want to map the local account in the second field, then click **OK**.

Once this policy is applied, users or services attempting to log in with the local mapped account must provide the Active Directory password for the account. For example, if you have mapped the local user caine to an Active Directory account that uses the password +shark1, the user logging in with the caine user name must provide the +shark1 password or authentication will fail.For more information about mapping local Linux, UNIX, or Mac OS X accounts to Active Directory accounts, see the *Administrator's Guide for Linux and UNIX* or the *Administrator's Guide for Mac*.

User's Initial Group ID

Use the group policy under **User's Initial Group ID** to specify the default group identifier (GID) to use for new users when you run the adupdate user add command.

Use FIPS 140-2 compliance algorithms

Use the **FIPS compliant algorithms for encryption, hashing and signing** group policy to specify the use of FIPS 140-2-compliant cryptographic algorithms for authentication protocols.

Basic requirements

Delinea supports FIPS 140-2 compliance for authentication using Kerberos and NTLM with the following requirements and caveats:

- FIPS mode is available on agent version 5.0.2 or later but only on supported operating systems. See the NIST validation entry for the Centrify FIPS mode for the current list of supported platforms.
- Domain controllers must be at Windows Server 2008 domain functional level, or later.
- The administrator must explicitly add the centrifydc_fips.xml or directly edit the administrative template to enable this policy.

  Note: Delinea recommends that you use the centrifydc_fips.xml template. When you do, the agent performs several checks before implementing the policy to confirm that your domain controller and joined computers meet the requirements.
- If multiple encryption types are specified only the AES128-CTS and AES256-CTS encryption type keys (with RSA for public key generation, DSA for digital signature generation and SHA1, SHA256, SHA384 or SHA512 for hashing) are generated and saved to the keytab file. However, if arcfour-hmac-md5 encryption is specified, the MD4Hash of the machine password will be generated and saved to the keytab file.

  Note: Which encryption types are used in each joined computer is controlled by a parameter set in each Linux, UNIX, or Mac OS X computer's configuration file. See the adclient.krb5.permitted.encryption.types description in the Notes section on Related Configuration Parameters for an explanation.
- Inter-realm keys for the AES128-CTS or AES256-CTS encryption types must be established between any trusted domains to enable Active Directory users to log on to a joined computer (see the ksetup utility to set up inter-realm keys).
- FIPS mode only allows NTLM pass-through authentication over SChannel. FIPS mode is not available for NTLM authentication over SMB or SMB2.
- In some environments, offline multi-factor authentication is not compatible with FIPS mode. See the *Multi-factor Authentication Quick Start Guide* for details about this restriction.

Enabling the Policy

To enforce FIPS 140-2 compliance, select the Computer Configuration › Policies› Centrify Settings › DirectControl Settings › **Use FIPS compliant algorithms for encryption, hashing, and signing** policy, open the properties, and select **Enabled**.

The policy takes effect after the next group policy update.

When you use the XML group policy template, the agent performs the following validation checks:

- It verifies that each joined computer is running a supported operating system.
- It verifies that each machine is joined to a domain at domain functional level 2008 or above. If the domain does not meet the domain functional level requirements, the agent issues the following warning:

  FIPS mode is supported only on domain with 2008 domain functional level or up.

  Enabling this policy with lower domain functional level may prevent adclient from working properly. Are you sure you want to enable this policy?

  Respond Yes to enable the policy regardless or No to abort. However, if the current domain functional level is inadequate or FIPS mode is not supported on the host platform, the agent does not restart when the policy is applied.

For all joined computers that pass, the agent is automatically stopped and restarted. After a successful restart, the adjoin, adleave, and adinfo commands run in FIPS mode immediately. If a joined computer is running an unsupported platform, the computer's configuration file is not updated and the agent is not restarted.

There are several restrictions and rules governing the use of FIPS mode. The following bullets summarize the policy:

- Pre-validated groups and users that use FIPS mode to log on when disconnected must have each user's Active Directory msDSSupportedEncryptionTypes

attribute set to use Kerberos AES 128- or 256-bit encryption. You can set this attribute in the users' accounts using Active Directory Users and Computers or ADSI Edit.
- The value of the corresponding Windows policy to use FIPS compliant algorithms has no effect on the Windows, Linux, UNIX, or Mac OS X computers managed through the Centrify Agent. You must use the Centrify policy to enable FIPS mode. The Centrify policy is only available when you add the centrifydc_fips.xml or centrifydc_fips.admx template (see Adding Centrify policies from XML files).

Related configuration parameters

The following centrifydc.conf configuration parameters affect FIPS operation. See the *Configuration and Tuning Reference Guide* for details about these parameters.

- fips.mode.enable: Enable FIPS mode on a per-computer basis. This group policy modifies the fips.mode.enable parameter in centrifydc.conf.
- adclient.krb5.clean.nonfips.enctypes: If FIPS mode is enabled and this configuration parameter is set to true, adclient scans the computer's keytab file and removes all non-AES encryption keys for service principal names (SPNs) during startup. The default is false.
- adclient.krb5.permitted.encryption.types: If FIPS mode is enabled, and if you include the arcfour-hmac-md5 encryption type in this configuration parameter, and if adclient.krb5.clean.nonfips.enctypes is true, adclient generates the MD4 hash for the computer password and saves it in the keytab file.

Account Prevalidation

Prevalidation enables specific users or the members of a specific group to access a Delinea-managed computer using their Active Directory credentials even if the following conditions would normally prevent them from logging on:

- The computer is disconnected from the network and unable to contact Active Directory to authenticate their identity.
- The user has not previously logged onto the computer.

Without prevalidation, only users who have previously logged on and had their password hashes stored in the local cache can be authenticated when the computer is disconnected from the network.

You can use the **Account Prevalidation** group policies to manage the users and groups who are authorized or denied access to disconnected computers.

Use the following group policies specify the users and groups that can be prevalidated:

- Specify Allowed Groups for Prevalidation
- Specify Allowed Users for Prevalidation

Use the following group policies specify the users and groups that cannot be prevalidated:

- Specify Denied Groups for Prevalidation
- Specify Denied Users for Prevalidation

Use the following group policies specify other prevalidation settings:

- Set Prevalidation Service Name
- Set Prevalidation Update Interval

Specify Allowed Groups for Prevalidation

Enable this policy and enter a comma-separated list of groups to prevalidate users in the specified groups for access Delinea-managed computers. To allow prevalidation for all users in the zone without any exceptions, you can enter all@zone in **Specify allowed groups for prevalidation**.

This group policy modifies the following setting in the agent configuration file:

adclient.prevalidate.allow.groups

Specify Allowed Users for Prevalidation

Enable this policy and enter a comma-separated list of users to prevalidate specific users for access Centrify-managed computers. This group policy modifies the following setting in the agent configuration file:

adclient.prevalidate.allow.users

Specify Denied Groups for Prevalidation

Enable this policy and enter a comma-separated list of groups that cannot be prevalidated for access Centrify-managed computers. If you allow any groups or users to be prevalidated, you can use this policy to define exceptions for any groups that should be prevented from prevalidation.

In most cases, you would use this policy to exclude a subset of users that are in a group that is a member of an allowed group. For example, you might want allow all users in the admins group to be prevalidated, except the users who are members of the nested outsource subgroup. To accomplish this, you would enable "Specify allowed groups for prevalidation" for the admins group, then use the "Specify denied groups for prevalidation" policy to deny access to users who are members of the outsource group.

This group policy modifies the following setting in the agent configuration file:

adclient.prevalidate.deny.groups

Specify Denied Users for Prevalidation

Enable this policy and enter a comma-separated list of users to prevent prevalidation of specific users for access Centrify-managed computers. If you allow any groups or users to be prevalidated, you can use this policy to define exceptions for any users who should be prevented from prevalidation. In most cases, you would use this policy to exclude a subset of users that are members of an allowed group.

This group policy modifies the following setting in the agent configuration file:

adclient.prevalidate.deny.users

Set Prevalidation Service Name

Enable this policy to specify the service name to use for prevalidated users and groups. You must use the name you specify in this parameter when you register the Service Principal Name (SPN) for a user or group with the setspn.exe utility. The default value is preval.

Setting the Service Principal Name for a User

For users or groups of users to be prevalidated, their accounts must be active accounts with permission to log on to the local computer and have a Service Principal Name (SPN) set in the form of:

```
preval/user
```

Where preval is the service name specified by the adclient.prevalidate.service parameter and username is the user logon name, which can be either of the following:

- the name part of the user's UPN, if the domain part matches the user's domain
- sAMAccountName, if the UPN is empty or the UPN's domain part is different from the user's domain

To enable prevalidation for a user, you can use the Windows setspn.exe utility to add a Service Principal Name for the user. For example, to register the Service Principal Name for the user kai@arcade.com using preval as the service name, you could type a command similar to the following in a Windows Command Prompt window:

```
setspn -A preval/kai kai
```

This setspn command registers the SPN in Active Directory for the preval service and the specified user account, for the Active Directory user kai. On the computers where this user is allowed to be prevalidated, the user can be authenticated without having logged on previously.

Setting the Service Principal Name for Group Members

If you are allowing prevalidation for an administrative group, you must register a Service Principal Name for each member of the group. For example, if you are allowing prevalidation for the admins group and this group has five members, you would use the setspn.exe utility to register a Service Principal Name for each of those members.

Set Prevalidation Update Interval

Enable this policy to specify the interval, in hours, for refreshing the credentials for prevalidated user and group accounts. The credentials for prevalidated users must be periodically refreshed to ensure they are in sync with Active Directory and that prevalidation will continue working after password changes.

The parameter value should be a positive integer. A value of 0 disables all prevalidation of users. The default is 8 hours.

This group policy modifies the adclient.prevalidate.interval setting in the agent configuration file.

Refreshing Prevalidated Credentials

Prevalidated credentials are periodically refreshed at the interval defined by the Set prevalidation update interval policy to ensure that prevalidation will continue working after password changes. In addition, the credentials for prevalidated users and groups are periodically retrieved from Active Directory whenever you do the following:

- Reboot the local computer.
- Start or restart the agent (adclient).
- Run the adflush command to clear the cache.
- Change a password from the local system.

Adclient Settings

Use the group policies under **Adclient Settings** to control the operation of the agent on managed computers.

Some of these policies are platform-specific policies that control whether the agent can automatically edit specific files on the local computer. In most cases, you should enable the policies that allow the agent to maintain configuration files automatically.

If you choose to not enable any of the platform-specific policies, you must manually edit the appropriate configuration files on individual computers. For example, if not configuring files automatically through a group policy, you must manually edit the /etc/nsswitch.conf and /etc/pam.d/system-auth or /etc/pam.d files to include adclient information or authentication through Active Directory will fail and you may disable login access entirely. For more information about updating configuration files manually, see *Customizing adclient configuration parameters* in the Configuration and Tuning Reference Guide.

> **Note**: Several Auto Zone group policies are located within the **Adclient Settings** node. For details about Auto Zone group policies, see Auto Zone Group Policies.

Add Attributes to Cached Objects

Use the following group policies to add specified Active Directory attributes to the local cache:

- Add attributes to cached user objects
- Add attributes to cached group objects
- Add attributes to cached computer objects

You can use the `adquery --dump` command to see which attributes are cached by default.

These policies modify the following parameters in the `centrifydc.conf` configuration file:

`adclient.custom.attributes.user`

`adclient.custom.attributes.group`

`adclient.custom.attributes.computer`

Auto Zone Group Policies

Use the **Auto Zone** group policies under **Adclient Settings** to set configuration parameters for all managed computers in the Auto Zone at once rather than configuring parameters for computers individually.

The Auto Zone group policies are defined in the centrifydc_settings.xml template file. These group policies and parameters have no effect on computers not joined to Auto Zone.

Auto Zone Default Shell

Set the default shell when joined to Auto Zone. The default value is:

- /bin/bash on Mac OS X and Linux computers
- /bin/sh on UNIX systems, including Solaris, HPUX, and AIX

This group policy modifies the auto.schema.shell parameter in the centrifydc.conf configuration file.

Delinea

Auto Zone Domain Prefix Overrides

Specify a unique prefix for a trusted domain. The Auto Zone algorithm combines the prefix with the lower 22 bits of each user or group relative identifier (RID) to create unique Linux, UNIX, or Mac OS X numeric user (UID) and group (GID) identifiers for each user and group in the forest and in any two-way trusted forests.

Ordinarily, you do not need to set this parameter because Delinea automatically generates the domain prefix from the user or group security identifier (SID). However, in a forest with a large number of domains, domain prefix conflicts are possible. When you join a computer to a domain, if Delinea detects any conflicting domain prefixes, the join fails with a warning message. You can then set a unique prefix for the conflicting domains.

To set this parameter, select **Enabled**, then click **Add**. Type a domain name and type a prefix or use the arrows to set a prefix number. The prefix must be in the range 0 - 511. Click **OK** to enter the prefix and domain. Add as many prefixes as you need, then click **OK** to close the group policy property page.

This group policy modifies the `auto.schema.domain.prefix` parameter in the agent configuration file.

Auto Zone Home Directory

Specify the default home directory. If you do not enable this policy, the default home directory will be based on the platform as follows:

- Mac OS X: /Users/%{user}
- Linux, HP-UX, and AIX: /home/%{user}
- Solaris: /export/home/%{user}

The variable %{user} specifies the logon name of the user. For example, if you specify /Users/%{user} and jsmith logs on to the Mac OS X computer, the home directory is set to /Users/jsmith.

This group policy modifies the auto.schema.homedir parameter in the agent configuration file.

Auto Zone Remote File Service (Mac OS X)

Specify the type of remote file service to use for the network home directory. The options are: SMB (default) and AFP. This group policy only applies to Mac OS X computers. When you type a path for the network home directory in Active Directory, it requires the format /server/share/path, but on Mac OS X computers, the format for mounting a network directory requires the remote file service type as part of the path/type/server/share/path. By identifying the remote file-service type, you can type the network path in the format required by Active Directory, and convert the path into the format required by Mac OS X computers.

This group policy modifies the auto.schema.remote.file.service parameter in the agent configuration file.

Generate New UID/GID using Apple Scheme In Auto Zone

Use the Apple algorithm to automatically generate user and group identifiers. The Apple algorithm for generating identifiers is based on the objectGuid attribute for the user or group object. The Centrify mechanism for automatically generating UIDs and GIDs is based on the security identifier for user or group objects. Both methods ensure a globally unique and consistent identifier for the user or group.

This group policy modifies the auto.schema.apple_scheme parameter in the agent configuration file.

Set User's Primary GID in Auto Zone

Specifies the group identifier (GID) to use as the default primary group for all users. If this policy is not configured, the primary GID for users in Auto Zone is set to one of the following platform-specific values:

- Mac: 20
- Linux, Solaris, HPUX, AIX: -1

If you enable this group policy, you must specify an integer from -1 to 2147483647. You cannot leave the GID field blank if you enable this group policy.

If you set this group policy to -1, the primary GID is generated according to the selected scheme:

- Apple scheme
- Relative identifier (RID)
- Active Directory value

This group policy modifies the `auto.schema.primary.gid` parameter in the agent configuration file.

Delinea

Specify AD Groups Allowed In Auto Zone

Specify the Active Directory groups that are included in the Auto Zone. By default, all Active Directory groups are included in the Auto Zone. When you enable this policy, only the specified groups are included in the Auto Zone and assigned a GID on the computer.

You can manually enter each group name separated by a comma, or click **List**, then **Add**, to browse for groups to add. If you manually add groups, use one of the following formats:

- SAM account name
- NTLM: DOMAIN\\sAMAccountName (also DOMAIN/sAMAccountName)
- UPN or sAMAccountName@domain
- Full DN: CN=commonName, …,DC=domain_component, DC=domain_component,…
- Canonical Name : domain.com/container1/cn

You can also specify the groups in a file.

Any groups listed may be domain local, global, or universal security groups. Distribution groups are not supported. If an Active Directory user specified in "Specify AD users allowed in Auto Zone" is a member of a group that is *not* specified in the current group policy, that group is ignored.

This group policy modifies the auto.schema.groups parameter in the agent configuration file.

Specify AD Users Allowed in Auto Zone

Specify the Active Directory users that are included in the Auto Zone and able to log in using their Active Directory account.

By default, all Active Directory users are included in the Auto Zone. When you enable this policy, only the specified users and members of the groups specified with the Specify Groups of AD Users Allowed in Auto Zone policy are included in the Auto Zone and able to log in using their Active Directory account.

You can manually enter each user name separated by a comma, or click **List**, then **Add**, to browse for users to add. If you manually add users, use one of the following formats:

- SAM account name
- NTLM: DOMAIN\\sAMAccountName (also DOMAIN/sAMAccountName)
- UPN or sAMAccountName@domain
- Full DN: CN=commonName, …,DC=domain_component, DC=domain_component,…
- Canonical Name : domain.com/container1/cn

You can also specify the users in a file.

This group policy modifies the auto.schema.allow.users parameter in the agent configuration file.

Delinea

Specify Groups of AD Users Allowed in Auto Zone

Specify the Active Directory users that are included in the Auto Zone by specifying the groups whose members should be included. By default, all Active Directory users are included in the Auto Zone. When you enable this policy, only the users listed for the Specify AD Users Allowed in Auto Zone policy and members of the listed groups (including members of nested groups under these groups and users' whose primary group are set to these groups) are included in the Auto Zone.

You can manually enter each group name separated by a comma, or click **List**, then **Add**, to browse for groups to add. If you manually add groups, use one of the following formats:

- SAM account name
- NTLM: DOMAIN\\sAMAccountName (also DOMAIN/sAMAccountName)
- UPN or sAMAccountName@domain
- Full DN: CN=commonName, ...,DC=domain_component, DC=domain_component,...
- Canonical Name : domain.com/container1/cn

You can also specify the groups in a file.

Any groups listed may be domain local, global, or universal security groups. Distribution groups are not supported.

This policy does not include the group in Active Directory Auto Zone, just the users in that group. This means that the group is *not* automatically assigned a GID. Use the Specify AD Groups Allowed in Auto Zone group policy to include a group in the Auto Zone and assign it a GID.

Auto Zone does not support one-way trusts. Therefore, any users in the group who belong to a domain that has a one-way trust relationship to the joined domain do not become valid users on the computer.

This group policy modifies the auto.schema.allow.groups parameter in the agent configuration file.

Configure /etc/nsswitch.conf (Solaris, HPUX, Linux)

Allow automatic editing of the Name Service Switch configuration (nsswitch.conf) file on **HP-UX**, **Solaris**, and **Linux** computers. This policy modifies the adclient.autoedit.nss setting in the agent configuration file.

Configure /etc/{pam.conf,pam.d} (AIX, Solaris, HPUX, Linux, Mac OS X)

Allow automatic editing of the PAM configuration (pam.conf file or pam.d directory) on **AIX, HP-UX**, **Solaris**, **Linux, and Mac OS X** computers. This policy modifies the adclient.autoedit.pam setting in the agent configuration file.

Configure /etc/security/user (AIX)

Allow automatic editing of the LAM user configuration files on **AIX** computers. This policy modifies the adclient.autoedit.user setting in the agent configuration file.

Configure /usr/lib/security/methods.cfg (AIX)

Allow automatic editing of the LAM methods.cfg files on **AIX** computers. This policy modifies the adclient.autoedit.methods setting in the agent configuration file.

Configure Directory Services (Apple OS/X)

Allow automatic editing of the Directory Service configuration on **Mac OS X** computers This policy modifies the adclient.autoedit.dsconfig setting in the agent configuration file.

Configure Dump Core Setting

Specify whether the agent should be allowed to dump core. The value you set for this group policy overrides the default `ulimit` setting. When you enable this group policy, select one of the following options from the drop down menu:

- never to specify that the agent never dump core.
- once to specify that the agent should dump core only when there is no existing core dump file. Note that this setting is not valid on Mac OS X computers. On Mac OS X, once behaves the same as always, which dumps core on every crash.
- always to specify that the agent dump core on every crash.

This policy modifies the `adclient.dumpcore` setting in the agent configuration file.

Disable Multi-Factor Authentication (MFA) on Delinea-Managed Computers

Enabling this policy disables multi-factor authentication on Delinea-managed computers. By default, this policy is “Not configured” which allows multi-factor authentication to be used if roles or rights are configured to require it.

This policy modifies the adclient.mfa.enabled setting in the agent configuration file.

Disable nscd Group and passwd Caching (Solaris, Linux)

Do not allow editing of the name service cache daemon configuration (nscd.conf) on **Solaris** and **Linux** computers. Note that selecting this policy disables rather than enables automatic editing of the file. This policy modifies the adclient.autoedit.nscd setting in the agent configuration file.

Disable pwgrd (HPUX)

Do not allow automatic editing of the password and group hashing and caching daemon (pwgrd) on **HP-UX** computers. Note that selecting this policy disables rather than enables automatic editing of the file. This policy modifies the adclient.autoedit.pwgrd setting in the agent configuration file.

Enable Core Dump Cleanup

Specify whether to delete old core dumps generated by the agent. By default, this policy is not configured, and core dumps generated by the agent will never be deleted. If you enable this group policy, agentgenerated core dumps are kept for the number of days that you specify. The default value is 30 days, but you can specify any number of days.

On Mac OS X, the default core dump location is /cores/. On most UNIX systems, the core dump location is the working directory of the current process. However, the core dump location can be customized on some platforms, including RHEL, Solaris, and AIX.

If the core dump location is inside /var/centrifydc and you enable this policy, all old core dumps are deleted without checking the process name first. If the core dump location is somewhere other than /var/centrifydc and you enable this policy, only the core dumps generated by the agent processes (for example, adclient, cdcwatch, and kcm) are deleted.

This policy does not modify the agent configuration file.

Enable Logon Hours Local Enforcement

Specify whether you want both Active Directory and the Centrify Agent to check for user logon hour restrictions, or just Active Directory. If you disable this policy, only Active Directory will check the user logon hour restrictions. By default, the configuration parameter set by this policy is set to `true`.

You might want to set this parameter to false if the user and Centrify Agent are in different time zones, and one time zone recognizes Daylight Savings Time, while the other does not. Otherwise, the user might not be able to log on at certain times.

This group policy modifies the `adclient.logonhours.local.enforcement` setting in the agent configuration file.

Encrypt adclient Cache Data

Specify to encrypt the local cache of Active Directory data. If you enable this policy, all of the Active Directory data stored in the cache is encrypted and the cache is flushed each time the agent starts up. If you disable or do not configure this policy, the cache is not encrypted and is not flushed when the agent starts up.

This group policy modifies the `adclient.cache.encrypt` setting in the agent configuration file.

Force Domains and Forests to be One-Way Trusted

Use the Force domains and forests to be one-way trusted group policy to specify a list of two-way trusted domains that need to be treated as one-way trusted domains. This is useful when two-way trusted domains are not accessible from UNIX machines, for example, they are behind a firewall. Configuring this parameter allows x-forest users to authenticate onto the trusting machines.

To set this group policy, select **Computer Configuration > Centrify Settings > DirectControl Settings > Adclient Settings > Force domains and forests to be one-way trusted**.

The default is an empty list.

Provide the following information for the group policy:

- A list of forests or domains to be treated as one-way trusted.

  Specify a list of two-way trusted forests, and domains that have two-way external trust relationship with the local domain, to be treated by DirectControl Agent as one-way trusted forests or domains.

This parameter is likely to be used together with the configuration parameters, **Specify NTLM authentication domains** and **Specify AD to NTLM domain mappings**, if these forests and domains are not accessible from UNIX machines.

- Use the group policy, **Specify NTLM authentication domains**, to specify the list of domains that use NTLM authentication instead of Kerberos authentication.
- Use the group policy, **Specify AD to NTLM domain mappings**, to map AD domains to NTLM domains.

Alternative to using this group policy, **Force domains and forests to be one-way trusted**, you can use the configuration parameter, adclient.one-way.x-forest.trust.force.

Force Password Salt Lookup from KDC

Force the Centrify Agent to look up the complete principal name, including the Kerberos realm used as the key salt, from the KDC. Enabling this policy is only required if you remove arcfour-hmac-md5 from the list of encryption types specified for the adclient.krb5.tkt.encryption.types parameter in agent configuration file and if you change a userPrincipalName attribute in Active Directory without changing the user's password.

Enabling this policy may cause "pre-auth required" warning messages to appear in the Active Directory event log.

This group policy modifies the adclient.force.salt.lookup setting in the agent configuration file.

Map /home to /User (Mac OS X)

Although this group policy is defined in the centrifydc_settings.xml file, not in the mac_settings.xml file, it applies to Mac OS X computers only. See the *Administrator's Guide for Mac* for a description of this policy.

Run adclient on all Processors

Specify whether to use all processors on a multi-processor system. By default, adclient uses all processors.

This policy modifies the adclient.use.all.cpus setting in the agent configuration file. This parameter is set to true by default. Disable this policy to set the parameter to false if adlcient becomes unstable.

Set Cache Cleanup Interval

Specify how often the agent should clean up the local cache. At each cleanup interval, the agent checks the cache for objects to be removed or expired, and at every 10th interval, the agent rebuilds local indexes. The value should be less than the values specified for the following parameters in the Centrify Agent configuration file:

adclient.cache.negative.lifetime adclient.cache.flush.interval adclient.cache.object.lifetime

The default cleanup interval is 10 minutes.

This group policy modifies the adclient.cache.cleanup.interval setting in the agent configuration file.

Set the Connector Refresh Interval

This policy controls how frequently connections to Centrify Connectors are refreshed. The refresh task is a background process that searches for and selects the nearest available connector to use for connectivity between the Active Directory forest and the identity platform service.

By default, the process runs every 8 hours. You can use this group policy to modify that interval. If the interval is set to 0, the refresh task will be suspended.

This group policy modifies the adclient.cloud.connector.refresh.interval parameter setting in the agent configuration file.

) [tags]: # (heartbeat,NIX) [priority]: # (46)

Set the Heartbeat Interval (

Use this policy to specify how often (in minutes) adclient will send an INFO message to the UNIX syslog.

By default, this policy is set to zero (0), which means that this task is disabled.

Set Maximum Number of Threads

Specify the maximum number of threads the agent will allocate for processing client requests. The value should be greater than or equal to the number of pre-allocated threads specified by the Set minimum number of threads policy. If you do not enable the policy, the default value is 20 threads.

This group policy modifies the adclient.clients.threads.max setting in the agent configuration file.

Set the Maximum Simultaneous Authentication Requests Allowed

This policy specifies the maximum number of identity platform authentication requests that can be processed simultaneously. The default is 10 simultaneous requests.

If you change this setting, you must restart the adclient process.

This group policy modifies the adclient.cloud.auth.token.max setting in the agent configuration file.

Set Minimum Number of Threads

Specify the number of threads the agent pre-allocates for processing client requests. The value must be an integer, zero or greater. If you set the value to zero, the agent processes requests sequentially. If you do not enable this policy, the default value is 4 threads.

This group policy modifies the adclient.clients.threads setting in the agent configuration file.

Specify Low Disk Space Interval

Specify how frequently the agent should check the disk space available for the local cache. The default interval checks the available disk space every 5 minutes. If the disk space available at any interval is less than the value you set for the Specify low disk space warning level policy, the agent will stop saving data in the local cache and will discard any new data until you free up enough disk space for it to resume saving data in the local cache.

The value must be an integer zero or greater. A value of zero disables checking for available disk space.

Keep in mind that the value you set for this policy can affect the recovery of a system after the agent stops writing data to the local cache. If you set the value to 0, the agent will not check for available disk space so it will not return to normal operation when disk space is freed up. In addition, setting value to 0 or to a long interval may cause the agent to consume too much of the disk for its local cache and make the computer unstable or unusable. Therefore, you should keep the interval for checking the available disk space relatively short. Keeping the interval short will also help to ensure that the agent resumes normal operation and saving data to its cache at the earliest opportunity.

This group policy modifies the adclient.disk.check.interval setting in the agent configuration file.

Specify Low Disk Space Warning Level

Generate a warning message when the disk space available for the local cache reaches a critical level. If you enable this policy, you also need to specify the threshold for available disk space that should trigger the warning message. By default, the warning is triggered if the free disk space reaches 51200 KB. Setting the Minimum Free Disk Space to 0 KB disables the display of a warning message.

If you enable the Specify low disk space interval policy, the agent will check the availability of free disk space at the interval specified. If the disk space available at any interval is less than the KB you set for the warning level, the agent stops saving data in the local cache. At the next interval when the available disk space exceeds the KB you set for this policy, the agent resumes normal operation and saving data to its cache.

Keep in mind that the value you set for this policy can affect the recovery of a system. The agent will only resume writing data to its local cache if there is more disk space available than what you have specified to generate the warning.

This group policy modifies the adclient.disk.check.free setting in the agent configuration file.

Specify a Per Machine (Random) Delay for Cache Refreshed Background Tasks

This group policy allows you to specify a per machine (random) delay, in minutes, for cache refreshed background tasks.

When there are more than one machines joined to the same domain and a number of those machines schedule background tasks to frequently access AD at the same time, the convergence of these activities causes a delay in AD. If you stagger these activities, you can avoid the convergence.

Once defined, scheduling background tasks calculates a random period of time within the interval and adds the same time to the delay of the tasks. If you change the interval setting, the period of time is recalculated. This only applies to newly scheduled background tasks.

The default setting is 0 and no delay. This policy modifies the queueable.random.delay.interval setting in the Centrify DirectControl configuration file.

Use the Legal Kerberos Type for Cache Encryption

Specify the type of encryption to use when encrypting the local cache. The encryption type you specify must be a type supported in the Kerberos environment. For example, Windows Server 2003 Kerberos supports the following cryptographic algorithms: RC4-HMAC, DES-CBC-CRC and DES-CBC-MD5.

This group policy is only used if the Encrypt adclient cache data policy is enabled. If Encrypt adclient cache data is not enabled, this policy is ignored.

This group policy modifies the adclient.cache.encryption.type setting in the agent configuration file.

addns Settings Group Policies

Use the group policies under **Addns Settings** to configure domain name service settings in the agent configuration file.

Enable addns Invoked by adclient

Enable whether adclient automatically launches the addns command. The addns command dynamically updates the DNS records on an Active Directory-based DNS server in environments where the DHCP server cannot update DNS records automatically.

In most cases, you do not need to use the addns command if a host's IP address is managed by a Windows-based DNS server and the host obtains its IP address from a Windows-based DHCP server because the DHCP server updates the DNS record for the host automatically.

If you are not using a Windows-based DNS server, you should use nsupdate or a similar command appropriate to the operating environment of the DNS server to update DNS records.

You can set the parameters of the addns command by specifying them in the Set Command Line Options Used by adclient group policy.

The default value for Mac OS X computers is True. The default value for all other platforms is False.

This group policy modifies the adclient.dynamic.dns.enabled parameter in the agent configuration file.

Set Command Line Options Used by adclient

Specify the parameters to use for the addns command if it is enabled by the Enable addns Invoked by adclient group policy. For example, the default setting is:

```
/usr/sbin/addns -U -m
```

The -U option creates or updates the IP address and domain name pointer (PTR) records in the DNS server for the local computer.

The -m option uses the local computer account's Active Directory credentials to establish a security context with the DNS server.

Note that computers that act as a gateway between networks may require you to specify the network adapter IP address in the addns command line. To ensure that you register the correct network address with the Active Directory DNS server, set adclient.dynamic.dns.command with a command line that uses the correct IP address for the network interface you want to use.

This group policy modifies the adclient.dynamic.dns.command parameter in the agent configuration file.

Set DNS Records Update Interval

Specify whether or not dynamic DNS records are periodically updated for this host and, if there are updates, the interval between updates. This interval value is defined in seconds and takes an integer of 0 or greater. If you set the value to 0, the DNS update feature will be disabled. Set the value to 1 or greater to specify the number of seconds between DNS update attempts.

The default for the is parameter is 0.

This group policy modifies the adclient.dynamic.dns.refresh.interval parameter in the agent configuration file.

Set Wait Response Interval for Update Requests

Specify the amount of time, in seconds, that the addns process waits for responses to its request for updates. The parameter value takes an integer of 0 or greater. The default value for this policy is 7 seconds.

This group policy modifies the addns.tcp.timeout parameter in the agent configuration file.

dzdo Settings

Use the group policies under **Dzdo Settings** to control the operation of dzdo.

Always Add Anchors to Regex in dzdo and dzcmds

Specifies whether you wan to add anchors automatically to the regular expressions you define as command rights and use in role definitions. This group policy helps to prevent matching unintended paths or commands if the regular expression pattern is not carefully set.

If you set this group policy to **Disabled**, you should carefully review all regular expressions used as command rights to identify all possible matches for the pattern defined.

This group policy modifies the dzdo.auto.anchors setting in the agent configuration file.

Enable Logging of Valid Command Execution in dzdo

Specify whether messages resulting from successful command execution are logged. Messages are written to the syslog auth facility or authpriv facility, typically located in /var/log/secure.

If you set this group policy to **Not configured** or **Enabled**, the dzdo program logs both valid and invalid command execution.

If you set this group policy to **Disabled**, information about only invalid command execution is logged.

This group policy modifies the dzdo.log_good setting in the agent configuration file.

Enable User Command Timeout

This group policy modifies the `dzdo.user.command.timeout` setting in the Centrify DirectControl configuration file. When this group policy is set to **Enabled**, the user may specify a timeout on the `dzdo` command line with a `-T` option. If the timeout expires before the command has exited, the command will be terminated. The default setting is disabled.

Force dzdo Re-Authentication when Relogin

Specify whether users must authenticate again with dzdo after logging out.

When a user authenticates with dzdo, a ticket is temporarily created that allows dzdo to run without re-authentication for a short period of time. If a user logs out, the ticket is reused when the user logs back in.

Enable this policy to remove the tickets when a user logs out. The user will be required to re-authenticate again when logging back in.

The default, when the policy is not set, is to not clear the tickets when users log out.

This group policy modifies the adclient.dzdo.clear.passwd.timestamp setting in the agent configuration file.

Force dzdo to Set HOME Environment Variable

Specify whether privileged commands run with dzdo commands should set the HOME environment variable to the home directory of the target user (which is root by default).

If you set this group policy to **Not configured** or **Disabled**, the dzdo program does not set the HOME environment variable.

If you set this group policy to **Enabled**, the dzdo program sets the HOME environment variable. Enabling this group policy effectively implies that the -H command line option should always used.

This group policy provides functionality equivalent to setting the always_set_home flag for configuring sudo operation.

This group policy modifies the dzdo.always_set_home setting in the agent configuration file.

Force dzdo to Set HOME Environment Variable when Runs with '-s' Option

Specify whether privileged commands run with dzdo using the -s command line option should set the HOME environment variable to the home directory of the target user (which is root by default).

If you set this group policy to **Not configured** or **Disabled**, the dzdo program does not set the HOME environment variable.

If you set this group policy to **Enabled**, the dzdo program sets the *HOME* environment variable.

This group policy provides functionality equivalent to setting the set_home flag for configuring sudo operation.

This group policy modifies the dzdo.set_home setting in the agent configuration file.

Force per tty Authentication in dzdo

Specify whether dzdo requires authentication once per tty rather than once per user.

If you set this group policy to **Not configured** or **Disabled**, authentication is required once per user. If you set this group policy to **Enabled**, authentication is required once per tty.

This group policy provides functionality equivalent to setting the tty_tickets flag for configuring sudo operation.

This group policy modifies the dzdo.tty_tickets setting in the agent configuration file.

Prompt Error Message if Command not Found by dzdo

Specify whether the dzdo program informs the user when it cannot find a command in the user's PATH.

If you set this group policy to **Not configured** or **Enabled**, the dzdo program displays an error statement indicating that the command could not be found in the user's PATH.

If you set this group policy to **Disabled**, dzdo is prevented from indicating whether a command was not allowed or simply not found.

This group policy provides functionality equivalent to setting the path_info flag for configuring the sudo operation.

This group policy modifies the dzdo.path_info setting in the agent configuration file.

Replace sudo by dzdo

Specify whether to replace sudo with dzdo.

Enable this policy to redirect sudo commands to dzdo. This policy creates a symbolic link between sudo and dzdo. When a user executes a sudo command, dzdo is executed instead. Role assignment settings for the user determine whether the user is allowed to execute the commands specified with sudo.

Be certain to set /usr/share/centrifydc/bin as the first search directory for the PATH variable if you enable this group policy.

This policy is only applicable if you are using zones. It is not applicable for computers that join Auto Zone.

Require dzdo Command Validation Check

Specify whether to enforce the validation check for dzdo privileged commands.

If you set this group policy to **Enabled**, privileged commands will run only after being validated by the dzdo validator. If a command fails validation, or if the dzdo validator does not exist, is not available, or is not trusted—for example because it is not owned by root or is group or world writeable—the command will not run.

If you set this group policy to **Not configured** or **Disabled**, no attempt is made to validate privileged commands, and the commands will run without validation.

This group policy modifies the dzdo.validator.required setting in the agent configuration file.

The dzdo validator is located and configured as described in Set dzdo Validator later in this section.

Require runas User for dzdo

Specify whether a user must explicitly identify the 'runas' user when executing a command with dzdo.

If you set this group policy to **Not configured** or **Enabled**, and a user executes a command with dzdo and does not explicitly identify the user or group to run as with the -u or -g option, adclient assumes that the command should be run as root. If the user is not authorized to run the command as root, dzdo fails to execute the command and issues an error message.

If you set this group policy to **Disabled** and a user executes a command with dzdo that does not explicitly identify the user or group to run as, adclient attempts to resolve the user. If the command defines a single runas user, dzdo executes the specified command and sends a message to the log file.

If the command defines multiple runas users, dzdo cannot resolve the user to run as and attempts to run the command as root. Because the user is not authorized to run the command as root, dzdo fails to execute the command and issues an error message.

In all cases, a user can execute a command successfully with dzdo by using the \-u option to explicitly identify the runas user. For example:

```
[u1@rh6]\$dzdo -u qa1 adinfo
```

This group policy modifies the dzdo.set.runas.explicit setting in the agent configuration file.

Require User is Logged in to a Real tty to Run dzdo

This group policy ensures a user is logged in to a valid tty to run dzdo. This policy modifies the dzdo.requiretty setting in the Centrify DirectControl configuration file. By default, this group policy is not configured, and you do not require a tty to run dzdo.

Set Directory to Store User Timestamp by dzdo

Specify the directory where dzdo stores the user's login timestamp files.

If you set this group policy to **Not configured** or **Disabled**, the default directory /var/run/dzdo is used.

If you set this group policy to **Enabled**, you can specify a directory of your choice.

This group policy provides functionality equivalent to setting the timestampdir flag for configuring sudo operation.

This group policy modifies the dzdo.timestampdir setting in the agent configuration file.

Set dzdo Authentication Timeout Interval

Specify the maximum number of minutes allowed between operations before prompting the user to re-enter a password.

If you set this group policy to **Not configured** or **Disabled**, the default timeout interval of five minutes is used. If you set this group policy to **Enabled**, you can specify a timeout interval of your choice.

You can set this parameter to zero (0) to always prompt for a password when users run privileged commands with dzdo. If you specify a value less than 0, the user's timestamp never expires.

This group policy provides functionality equivalent to setting the timestamp_timeout flag for configuring sudo operation.

This group policy modifies the dzdo.timestamp_timeout setting in the agent configuration file.

Set dzdo Password Prompt Timeout Interval

Specify the number of minutes before the dzdo password prompt times out.

If you set this group policy to **Not configured** or **Disabled**, the default timeout value of five minutes is used. If you set this group policy to **Enabled**, you can specify a timeout value of your choice.

You can set this parameter to zero (0) to have the password prompt never timeout.

This group policy provides functionality equivalent to setting the passwd_timeout flag for configuring sudo operation.

This group policy modifies the dzdo.passwd_timeout setting in the agent configuration file.

Set dzdo Validator

Specify the full path of the dzdo validator. The settings in this group policy are used only when the Require dzdo Command Validation Check group policy is enabled.

The dzdo validator is a script that runs synchronously under the user's Active Directory name. If the Require dzdo Command Validation Check group policy is enabled, the dzdo validator runs when users attempt to execute dzdo commands. Command attempts that pass validation are allowed to run. Command attempts that fail validation are not allowed to run.

The default location of the dzdo validator is /usr/share/centrifydc/sbin/dzcheck. If you set this group policy to **Not configured** or **Disabled**, the validator located in this default location is used.

If you set this group policy to **Enabled**, the dzdo validator that you specify is used.

Note that the Server Suite distribution package does not include a dzcheck script. Instead, a sample validator, /usr/share/centrifydc/sbin/dzcheck.sample, is provided for reference. To configure and enable the dzdo validator, modify the sample script or create a new script, then place that script in the default location (/usr/share/centrifydc/sbin/dzcheck) or use a location and script name of your choice that you specify in this group policy.

You do not need to create a dzcheck script to use dzdo. You only need to create a script if you want to modify dzdo behavior so that validation occurs when dzdo commands attempt to run.

This group policy modifies the dzdo.validator setting in the agent configuration file. For more information about configuring the dzdo validator, see the "dzdo.validator" section in the *Configuration and Tuning Reference Guide*.

Set Environment Variables to be Preserved by dzdo

Specify the default list of environment variables to preserve in the user's environment. This group policy applies only if you have selected the **Reset environment variables** option for the command in Access Manager.

If you set this group policy to **Not configured** or **Disabled**, the default list of variables displayed when you run the `dzdo -V` command as root is preserved.

If you set this group policy to **Enabled**, you can specify variables to preserve in addition to the default list of variables. Variables that you specify must be formatted as a comma-separated list. For example:

```
COLORS,DISPLAY,HOME,HOSTNAME,KRB5CCNAME, LS_COLORS,MAIL,PATH,PS1,PS2,TZ,XAUTHORITY,XAUTHORIZATION
```

This group policy provides functionality equivalent to setting the `env_keep` flag for configuring `sudo` operation.

This group policy modifies the `dzdo.env_keep` setting in the agent configuration file.

Set Environment Variables to be Removed by dzdo

Specify the default list of environment variables to be removed from the user's environment. This group policy applies only if you have selected the **Remove unsafe environment variables** option for the command in Access Manager.

If you set this group policy to **Not configured** or **Disabled**, the default list of variables displayed when you run the `dzdo -V` command as root is removed.

If you set this group policy to **Enabled**, you can specify variables to remove in addition to the default list of variables. Variables that you specify must be formatted as a comma-separated list. For example:

```
IFS,CDPATH,LOCALDOMAIN,RES_OPTIONS,HOSTALIASES, NLSPATH,PATH_LOCALE,LD_\*
```

This group policy provides functionality equivalent to setting the `env_delete` flag for configuring `sudo` operation.

This group policy modifies the `dzdo.env_delete` setting in the agent configuration file.

Set Environment Variables to be Removed by dzdo with Characters % or /

Specify the list of environment variables that should be checked for percent (%) or slash (/) special characters If there are environment variable values containing the special characters, dzdo removes those variables from the user's environment. Variables with % or / characters are removed regardless of whether you have selected the **Reset environment variables** option for the command in Access Manager.

If you set this group policy to **Not configured** or **Disabled**, the default list of variables displayed when you run the `dzdo -V` command as root is checked for special characters.

If you set this group policy to **Enabled**, you can specify variables to check for special characters in addition to the default list of variables. Variables that you specify must be formatted as a comma-separated list. For example:

```
COLORTERM,LANG,LANGUAGE,LC_\*,LINGUAS,TERM
```

This group policy provides functionality equivalent to setting the `env_reset` flag for configuring `sudo` operation.

This group policy modifies the `dzdo.env_check setting` in the agent configuration file.

Set Error Message when Failed to Authenticate in dzdo

Specify the message that is displayed if a user enters an incorrect password.

If you set this group policy to **Not configured** or **Disabled**, the default message "Sorry, try again" is used. If you set this group policy to **Enabled**, you can specify a message of your choice. The message can be any text string enclosed by quotation marks. For example:

"The password provided is not valid."

This group policy provides functionality equivalent to setting the badpass_message flag for configuring sudo operation.

This group policy modifies the dzdo.badpass_message setting in the agent configuration file.

Set Lecture Shown by dzdo Before Password Prompt

Specify the full path to a file containing the warning message that is displayed about using dzdo before displaying the password prompt.

If you set this group policy to **Not configured** or **Disabled**, a default message is used. If you set this group policy to **Enabled**, you can specify a file containing a message of your choice. You must specify the full path to the file. For example, to use a custom message located in the file dzdo_warning:

```
/etc/custom/dzdo_warning
```

This group policy provides functionality equivalent to setting the lecture_file flag for configuring sudo operation.

This group policy modifies the dzdo.lecture_file setting in the agent configuration file.

Set Password Prompt for Target User Password in dzdo

Specify the password prompt displayed when running privileged commands. This group policy serves the same function as the `dzdo -p` command.

If you set this group policy to **Not configured** or **Disabled**, the default prompt `[dzdo] password for *%p*:` where `*%p*` is *root* unless specified otherwise.

If you set this group policy to **Enabled**, you can specify a prompt of your choice. You can use the following escapes in the prompt:

%u —Expands to the invoking user's login name.

%U—Expands to the login name of the user the command will be run as. If not specified, defaults to root.

%h—Expands to the local hostname without the domain name.

%H—Expands to the local hostname including the domain name.

%p—Expands to the user whose password is asked for.

%%—Collapses to a single % character.

This group policy modifies the `dzdo.passprompt` setting in the agent configuration file.

Set Paths for Command Searching in dzdo

Specify the search path for the dzdo program to use to look for commands and scripts that require privileges to run.

If you set this group policy to **Not Configured** or **Disabled**, no search path is set (that is, there is no default value). If you set this group policy to **Enabled**, you can specify a list of directories for the dzdo program to search for commands and scripts. The dzdo program will search in the specified directories no matter which path the command rights are configured to use in the Access Manager **System search path** option.

If command paths are configured in Access Manager using the **System search path** option and this group policy is **Disabled** or **Not Configured**, the following actions take place:

- The current user's path is used to search for the commands.
- Only the commands located under the System path are allowed to execute.

The search path that you specify can be a list of directories or the name of a file that contains the list of directories. For example, you can specify a file that contains the directories to search using the file: keyword and a file location:

```
file:/etc/centrifydc/customized_dzdo_directories
```

If you specify a file name, you should ensure that the file is owned by root and is not accessible to any other users.

This group policy modifies the dzdo.search_path setting in the agent configuration file.

Set Secure Paths for Command Execution in dzdo

Specify the path for the dzdo program to use when executing commands and scripts that require privileges to run.

If you set this group policy to **Not Configured** or **Disabled**, no specific path is set (that is, there is no default value). If you set this group policy to **Enabled**, you can specify the directory that dzdo uses. The dzdo program will execute only the commands and scripts that are located in the directory that you specify.

The path that you specify can be a list of directories or the name of a file that contains the list of directories. For example, you can specify a file that contains the directories to search using the file: keyword and a file location:

```
file:/etc/centrifydc/customized_dzdo_directories
```

Within the file, lines should contain paths separated by colons. For example, a file specifying two paths might look this this:

```
/etc/centrifydc/reports/exec_report_cmds:/usr/sbin/ora_cmds
```

If you specify a file name, you should ensure the file is owned by root and not accessible to any other users.

Setting this group policy and the Set paths for Command Searching in dzdo group policy to the same path is equivalent to setting the secure_path parameter in the sudoers configuration file.

This group policy modifies the dzdo.secure_path setting in the agent configuration file.

Show Lecture by dzdo Before Password Prompt

Specify whether dzdo displays a warning message about using dzdo before displaying the password prompt.

If you set this group policy to **Not configured** or **Disabled**, the message defined in the Set Lecture Shown by dzdo Before Password Prompt group policy (or in dzdo.lecture_file) is displayed one time.

If you set this group policy to **Enabled**, you can specify whether and how often the message is displayed. The values that you can specify are:

- once—Display the warning message only the first time the command is run.
- never—Never display a warning message.
- always—Display the warning message every time the program is invoked.

This group policy provides functionality equivalent to setting the lecture flag for configuring sudo operation.

This group policy modifies the dzdo.lecture setting in the agent configuration file.

Use realpath to canonicalize Command Paths in dzdo

Specify whether dzdo uses command paths resolved by realpath when searching for commands.

If you set this group policy to **Not configured** or **Disabled**, realpath is not used to resolve command paths. If you set this group policy to **Enabled**, realpath is used to expand all symbolic links and resolve references to:

/./

/../

extra **/** characters

This group policy modifies the dzdo.use.realpath setting in the agent configuration file.

Set the Type of Time Stamp Record

The privilege elevation service uses per-user timestamp files for credential caching. You can use this group policy to specify the type of timestamp record for the service to use.

If you set this group policy to **Not configured** or **Disabled**, the service uses the `tty` time stamp record type.

If you set this group policy to **Enabled**, you can set this group policy to any of the following values:

__global__: A single time stamp record is used for all of a user's login sessions, regardless of the terminal or parent process ID.

__ppid__: A single time stamp record is used for all processes with the same parent process ID (usually the shell). Commands run from the same shell (or other common parent process) will not require a password for dzdo.timestamp_timeout minutes (5 by default). Commands run by way of sudo with a different parent process ID, for example from a shell script, will be authenticated separately.

__tty__: One time stamp record is used for each terminal, which means that a user's login sessions are authenticated separately. If no terminal is present, the behavior is the same as ppid. Commands run from the same terminal will not require a password for dzdo.timestamp_timeout minutes (5 by default).

This group policy modifies the `dzdo.timestamp_type` setting in the agent configuration file.

Group Policy Settings

Use the group policies under **Group Policy Settings** to manage the Centrify group policy mapping programs.

Enable User Group Policy

Specify whether to enable user-based group policies. If you enable this policy, user-based group policies are enabled. If you explicitly disable this group policy, user-based policies are disabled.

If you do not set this policy, the default is to enable user-based policies on Mac OS X machines and disable user-based policies on all other Linux and UNIX based computers.

When this policy is **Disabled**, all user configuration Software Settings and user configuration Windows Settings group policies set for computers in Centrify zones are not applied. You must enable this policy if you want to use any Software Settings, Windows Settings, or Centrify Settings group policies on computers in a Centrify zone.

User configuration group policies enabled in a child organizational unit do NOT apply to users logging in to computers in the child organizational unit who are not in that organizational unit (for example, they are in the parent organizational unit only). See Applying Policies in Nested Organizational Units if you need to have different user configuration policies at different levels in the organizational unit hierarchy.

This group policy modifies the `gp.disable.user` setting in the agent configuration file.

Group Policy Commands Environment Variable List Running as Root

This policy specifies the list of environment variables that are exported to the environment so that a root user can run Group Policy commands. Use the forward slash "/" to specify environment variables and their values. Use a space to separate multiple name/value pairs. If an environment variable contains spaces, enclose the value in quotes or add a backslash "\" before the space.

You can also specify other environment variables as a value.

Set Group Policy Mapper Execution Timeout

Specify the maximum amount of time, in seconds, to allow for a group policy mapper program to run before the process is stopped.

This group policy modifies the gp.mappers.timeout setting in the agent configuration file.

Set Machine Group Policy Mapper List

Specify the list of mapper programs to run for computer-based policies.

You can use an asterisk (*) as a wild card to match a set of program names. For example, you can specify `a*` to match all programs with names that start with the letter `a`.

You can use an exclamation point (!) with a program name to exclude a program from the list. For example, you can specify `!mysample` to prevent the mapping program `mysample` from running.

This group policy modifies the `gp.mappers.machine` setting in the agent configuration file.

Set User Group Policy Mapper List

Specify the list of mapper programs to run for user policies.

You can use an asterisk (*) as a wild card to match a set of program names. For example, you can specify `a*` to match all programs with names that start with the letter `a`.

You can use an exclamation point (!) with a program name to exclude a program from the list. For example, you can specify `!mysample` to prevent the mapping program `mysample` from running.

This group policy modifies the `gp.mappers.user` setting in the agent configuration file.

Set Total Group Policy Mappers Execution Timeout

Specify the maximum amount of time, in seconds, to allow for all group policy mapper programs to run before stopping all mapper processes.

This group policy modifies the gp.mappers.timeout.all setting in the agent configuration file.

User Group Policy Commands Run as User

This policy specifies whether to run user group policy commands as the current user. The default for this parameter is false, which means that the service runs these commands as administrator/root.

The RunCommand.pl script reads this policy setting and if the policy s set to true, then it runs commands as follows:

```
su - \$args -\>user() -c \$command
```

Use User Credential to Retrieve User Policy

Use this group policy to distinguish whether to use user credentials instead of machine credentials to retrieve user policy. By default, machine credentials are used to retrieve user policy. However, if a computer object does not have permission to access user group policy objects, user policy will not be applied.

If you enable this group policy, user credentials are used to retrieve user policy.

This group policy modifies the gp.use.user.credential.for.user.policy setting in the agent configuration file.

Kerberos Settings

Use the group policies under **Kerberos Settings** to manage the Kerberos configuration.

Allow PAM to Create User Kerberos Credential Cache

Use this policy to specify whether PAM creates the Kerberos user credential cache.

If this group policy is **Enabled** or **Not Configured**, a Kerberos user credential cache is created. The Kerberos user credential cache can be file-based or it can be a KCM in-memory cache, depending on the krb5.cache.type setting in /etc/centrifydc/centrifydc.conf.

If this group policy is disabled, the Kerberos user credential cache is not created, and any attempt to perform an SSO operation will fail.

This group policy modifies the pam.auth.create.krb5.cache setting in the agent configuration file.

Allow Weak Encryption Types for Kerberos Authentication

Use this group policy to specify whether to allow weak encryption types for Kerberos authentication.

By default (not configured), this policy allows the weak encryption types specified in the configuration parameters adclient.krb5.permitted.encryption.types and adclient.krb5.tkt.encryption.types.

These encryption types include:

- des-cdc-crc
- des-cbc-md4
- dec-cbc-md5
- dec-cbc-raw
- des3-cbc-raw
- des-hmac-sha1
- arcfour-hmac-exp
- rc4-hmac-exp
- arcfour-hmac-md5-exp

If you disable this policy, the above encryption types will not be supported. Note that setting this policy to disabled may cause authentication failures in existing Kerberos environments that do not support strong cryptography. Users in these environments should leave this policy set to **Not Configured** or **Enabled** until their environment adopts stronger cyphers.

This policy modifies the adclient.krb5.allow_weak_crypto parameter in the agent configuration file.

Alternative Location for Credential Cache Directory

Use this policy to control the adclient.krb5.ccache.dir configuration parameter. For details, see *Configuration and Tuning Reference Guide*.

The adclient.krb5.ccache.dir parameter specifies the directory where Kerberos ccache files are stored when krb5.cache.type is FILE.

This is useful when Kerberos applications in Docker containers use the Kerberos cache files. This parameter, in conjunction with adclient.krb5.ccache.dir.secure.usable.check enables volume bind mapping so that Kerberos cache files in the host OS are available to the Docker containers.

Alternative Location for User .k5login Files

Use this policy to specify an alternative location for user .k5login files.

If specified, this string value will be used for the k5login_directory in the [libdefaults] stanza in krb5.conf and the user's .k5login file will be named as \<k5login_directory\>/\<unix_name\>.

For security reasons the specified directory should be owned by root and writeable by root only. If the directory does not exist, adclient will create it.

This group policy modifies the krb5.conf.k5login.directory setting in the agent configuration file.

Disable Kerberos Built-in ccselect Plugins

Use this policy to specify whether adclient should disable the Kerberos built-in ccselect plugins.

If this group policy is **Enabled** or **Not Configured**, adclient will disable all ccselect built-in plugins in the **plugins** section of the krb5.conf file when the group policy, Manage Kerberos Configuration, is enabled.

If this group policy is set to **Disabled**, the ccselect plugins will **not** be disabled.

This group policy modifies the krb5.conf.plugins.ccselect.disable configuration parameter in the agent configuration file.

Enable Kerberos Clients to Correct Time Difference

Enable Kerberos to automatically correct for a time difference between the system clock and the clock used by the KDC. You only need to enable this group policy if your system clock is drifting and the system is not using NTP and adclient SNTP settings.

This group policy modifies the krb5.use.kdc.timesync setting in the agent configuration file.

Force Kerberos to Only use TCP

Force all Kerberos requests to use TCP rather than UDP.

This group policy modifies the krb5.forcetcp setting in the agent configuration file.

Generate the Forwardable Tickets

Specify whether you want the Centrify Agent to create forwardable Kerberos user tickets. Creating a forwardable ticket allows a user's logon ticket to be sent to another computer and used to access to additional systems and resources.

If you select **Enabled** for this group policy, service tickets can be forwarded from one service or resource to another. If you do not want tickets to be forwarded, you can uncheck this option to prevent the agent from creating forwardable tickets.

This group policy modifies the krb5.forwardable.user.tickets setting in the agent configuration file.

Generate Kerberos Version Numbers for Windows 2000

Kerberos Version Numbers (kvno), allow tickets issued with a computer's previous key to be decrypted even when the ticket was issued before the computer changed it's password, but presented afterwords.

Windows 2000 does not support these kvnos, but you can enable this policy to generate version numbers that work with Windows 2000.

However, this feature requires Centrify's Kerberos libraries so older Kerberos applications may fail to understand the generated Kerberos version numbers. You can disable this policy to support older applications with the knowledge that the race condition just described may cause authentication failures.

This group policy modifies the krb5.generate.kvno setting in the agent configuration file.

Manage Kerberos Configuration

Indicate whether you want the Centrify Agent to automatically manage the Kerberos configuration files.

This group policy modifies the `adclient.krb5.autoedit` setting in the agent configuration file.

Renew Credentials Automatically

Specify whether to automatically reissue user credentials when they expire. If you enable this group policy, the Centrify Agent keeps a hash of the user's password in memory indefinitely. If you do not enable this policy, or if you explicitly disable it, a user's credentials periodically expire and the user must be reauthenticated by re-entering a valid password.

If you enable this policy, user credentials are automatically reissued, as needed, as long as the `adclient` process continues to run even if the computer is disconnected from Active Directory. If you stop or restart `adclient`, however, the user's password hash is removed from memory. After stopping or restarting `adclient`, users must be re-authenticated by logging on with a valid user name and password.

The default value is `false`.

This group policy modifies the `krb5.cache.infinite.renewal` setting in the agent configuration file.

Set Configuration Update Interval

Specify how frequently, in hours, the Centrify Agent should update the Kerberos configuration files.

This group policy modifies the `krb5.config.update` setting in the agent configuration file.

Set Kerberos UDP Preference Limit

Specify the maximum size packet that the Kerberos libraries will attempt to send over a UDP connection before retrying with TCP. If the packet size is larger than this value, only TCP will be tried. If the value is set to 1, TCP will always be used. The hard UDP limit is 32700. If you enter a value larger than this, the value is reset to 32700 when you apply the policy.

This policy only takes effect if the policy Force Kerberos to only use TCP is not configured or is disabled (the configuration parameter krb5.forcetcp is set to false).

If Force Kerberos to only use TCP is enabled and the agent is managing the krb5.conf file, it will set udp_preference_limit = 1, so that the Kerberos libraries will always use TCP.

If you do not enable this group policy, the default value is 1465.

This group policy modifies the krb5.udp.preference.limit setting in the agent configuration file.

Set Credential Renewal Interval

Specify how frequently, in hours, Kerberos credentials are renewed. A value of 0 disables renewal completely.

This group policy modifies the krb5.cache.renew.interval setting in the agent configuration file.

Set Password Change Interval

Specify how frequently, in days, the Centrify Agent should change the computer account password in Active Directory.

This group policy modifies the adclient.krb5.password.change.interval setting in the agent configuration file.

Set Password Change Verification Interval

Specify the interval, in seconds, that adkeytab waits between computer password change verification attempts.

This group policy modifies the adclient.krb5.password.change.verify.interval setting in the agent configuration file.

The default setting is 300 seconds (5 minutes).

Set Password Change Verification Attempts

Specify the number of times that adkeytab attempts to verify password changes after an initial, failed attempt.

Some environments, such as those using a read-only domain controller (RODC), can experience replication delays that may prevent Kerberos password changes to be verified through adclient. As a result of this delay, the new password may not be saved to the keytab file.

Increasing the number of verification attempts can address replication delays that may result from having a read-only domain controller.

This group policy modifies the adclient.krb5.password.change.verify.retries setting in the agent configuration file.

The default setting is 0, which means that adkeytab does not attempt additional password verification attempts after the initial failure.

Specify Credential Cache Type for AD Users

Specify the type of Kerberos credential cache that adclient will create when an Active Directory user logs in. You can specify a file-based or in-memory-based credential cache.

> **Note**: The use of in-memory credential caches is not supported on Mac OS X computers, therefore applying this group policy setting to a Mac OS X computer has no effect.

To specify the type of cache to create, click **Enabled**, then select the type of cache from **Kerberos credential cache type**.

If you select **File-based credential cache**, the Centrify Agent creates a file-based credential cache for each Active Directory user in /tmp when the user logs in. A file-based credential cache persists until the file is deleted.

If you select **In-memory credential cache provided by Centrify-KCM service**, the Centrify Agent creates an in-memory credential cache for each Active Directory user when the user logs in. The Centrify-KCM service, run as root, manages in-memory credential caches. When the adclient process starts up, if the policy is configured for an in-memory credential cache, adclient starts the KCM service. If you change the setting from file-based to in-memory while adclient is running, adclient starts the KCM service the next time it is forced to reload configuration parameters, for example, if you run the adgpupdate command to update group policy settings, or if a user opens a new session.

Setting this parameter affects new users only — not users who have already logged in. For example, if you change from a file-based, to an in-memory credential cache, Direct Control will continue to use the file-based credential cache for any user who was logged in at the time of the change. If a logged in user opens a new session, or a new user logs in, the agent will use an in-memory cache for them.

An in-memory credential cache ends as soon as the Centrify-KCM service is stopped.

This group policy modifies the krb5.cache.type setting in the agent configuration file.

Specify Groups to Infinitely Renew Kerberos Credentials

Specify a list of Active Directory groups whose members' Kerberos credentials require infinite renewal even after the users have logged out. Groups that you specify must be Active Directory groups, but do not need to be zone enabled. However, only zone enabled users in a group will have their credentials automatically renewed.

If this group policy is **Enabled**, group member's credentials are renewed automatically. You must use the following format to specify groups when you enable this group policy:

sAMAccountName@domain

For example:

test_group_sam@example.com

By default, this group policy is disabled.

This group policy modifies the krb5.cache.infinite.renewal.batch.groups setting in the agent configuration file.

Specify Maximum Kerberos Credential Cache Lifetime

Specify whether adclient deletes credentials from the Kerberos cache if they are the specified number of days old.

If this group policy is **Enabled**, the credentials will be cleared for all users whether or not they are logged on, have active processes running, or are specified in the following group policy lists:

- Specify Groups to Infinitely Renew Kerberos Credentials
- Specify Users to Infinitely Renew Kerberos Credentials

You can configure this group policy by enabling it and setting the value to the age of the credential cache to be cleared, in days.

The default value for the group policy is 0 days, which means that this group policy does not clear any credential caches.

This group policy modifies the krb5.cache.clean.force.max setting in the agent configuration file.

Specify Users to Infinitely Renew Kerberos Credentials

Specify a list of users whose Kerberos credentials require infinite renewal even after the users have logged out. Users that you specify must be zone enabled (that is, mapped users are not supported). If this group policy is enabled, user credentials are renewed automatically.

You can use any of the following formats to specify user names:

unixName

userPrincipleName

sAMAccountName

sAMAccountName@domain

For example:

test_user

test_user@example.com

test_user_sam

test_user_sam@example.com

By default, this group policy is disabled.

This group policy modifies the krb5.cache.infinite.renewal.batch.users setting in the agent configuration file.

Specify Whether CDC k5login Module Should Ignore .k5login for SSO

Specify whether the k5login module should ignore .k5login for SSO.

The default value is false.

This group policy modifies the krb5.sso.ignore.k5login setting in the agent configuration file.

Specify Whether Kerberos PAC Checksum Validation Should be Done

This group policy specifies whether or not to verify that the user's PAC (Privilege Authorization Certificate) information is from a trusted KDC (Key Distribution Center) so as to prevent what's referred to as a "silver ticket" attack.

When performing credential verification, a service ticket is fetched for the local system. After the credential is verified, the local system uses the PAC information in the service ticket.

This group policy takes effect when the policy is enabled or when DirectControl is using the user's PAC from a service ticket. This setting does not apply to retrieving the PAC by way of the S4U2Self protocol.

There are 3 possible values for this policy:

- disabled (default): NO PAC validation will be done at all.
- enabled: If PAC Validation fails, the PAC information is used and the user login is allowed.
- enforced: If PAC Validation fails, the PAC information is discarded and the user login is denied.

Setting this group policy to enabled or enforced will have significant impact on the user login and user's group fetch performance.

Strictly Enforce Default Encryption Types

This parameter specifies if DirectControl should add or replace the default encryption types listed in the settings, default_tgs_enctypes and default_tkt_enctypes in krb5.conf with the types specified in the setting adclient.krb5.tkt.encryption.types in centrifydc.conf.

- When this group policy is not set (default) — No change in behavior. It means DirectControl adds any additional encryption types.

  Default encryption types from centrifydc.conf are added, if they were not already listed. Other items that were already in default_tgs_enctypes and default_tkt_enctypes are left alone and not removed.

- When this group policy is set — DirectControl replaces the encryption types listed in the settings, default_tgs_enctypes and default_tkt_enctypes in krb5.conf to match exactly with the encryption types listed in the setting, adclient.krb5.tkt.encryption.types in centrifydc.conf.

  Default encryption types from centrifydc.conf are added, if they were not already listed. Other items that were already in default_tgs_enctypes and default_tkt_enctypes, and not in centrifydc.conf, are removed.

This group policy is set as follows: **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings > Control if strictly enforce the encTypes**.

Strictly Enforce Permitted Encryption Types

This parameter specifies if DirectControl should add or replace the permitted encryption types listed in the setting, permitted_enctypes in krb5.conf with the types specified in the setting, adclient.krb5.permitted.encryption.types in centrifydc.conf.

- When this group policy is not set (default) — No change in behavior. it means DirectControl adds any additional encryption types.

  Permitted encryption types from centrifydc.conf are added, if they were not already listed. Other items that were already in permitted_enctypes are left alone and not removed.

- When this group policy is set — DirectControl replaces the setting, permitted_enctypes in krb5.conf to match exactly with encryption types listed in the setting, adclient.krb5.permitted.encryption.types in centrifydc.conf.

  Permitted encryption types from centrifydc.conf are added, if they were not already listed. Other items that were already in permitted_enctypes, and not in centrifydc.conf, are removed.

This group policy is set as follows: **Computer Configuration > Centrify Settings > DirectControl Settings > Kerberos Settings > Control if strictly enforce the permitted_encTypes**.

Use DNS to Lookup KDC

Allow the agent to use DNS to locate the Kerberos Key Distribution Center (KDC).

This group policy modifies the krb5.use.dns.lookup.kdc setting in the agent configuration file.

Use DNS to Lookup Realms

Allow the agent to use DNS to locate Kerberos realms.

This group policy modifies the krb5.use.dns.lookup.realm setting in the agent configuration file.

Local Account Management Settings

Use the group policies under **Local Account Management** to control whether local accounts are managed by the agent, and other aspects of local account management by the agent.

Enable Local Account Management Feature

Specify whether the agent manages local users and groups on the computer where the agent is installed.

When this group policy is **Enabled**:

- The agent gets the local user and local group profiles from the zone, and updates the local password and local group files using the information defined in the zone.
- You can view and manage local users and groups in Access Manager as described in the *Administrator's Guide for Linux and UNIX*.

By default, this group policy is disabled (unless you upgraded from a Server Suite release in which it was enabled), and the agent does not manage local users and groups.

This group policy modifies the adclient.local.account.manage setting in the agent configuration file.

Notification Command Line

Define a command to process changes to local account profiles after the agent synchronizes local user and group profiles with profiles defined in a zone.

For example, if new local users are added, removed, or have their enabled/disabled status changed locally, the command that you define in this policy is executed. Typical activities that this command might perform include setting the password for new or updated local accounts, or notifying password vault about changes to local accounts and defining actions to take regarding those accounts.

When this policy is **Enabled**, the agent invokes the defined command in another process and passes a comma separated UNIX name list to the command for further processing.

By default, this policy is not configured (that is, no command is defined).

This group policy modifies the adclient.local.account.notification.cli setting in the agent configuration file.

This policy takes effect only when local account management is enabled through the **Enable local account management feature** group policy, or through the adclient.local.account.manage configuration parameter.

Logging Settings

Use the group policies under **Logging Settings** to control the following aspects of a computer's logging facilities:

- Set Adclient Audit Logging Facility
- Set General Audit Logging Facility
- Set Log Message Queue Size
- Set NIS Audit Logging Facility

Set adclient Audit Logging Facility

Specify the syslog facility to use for logging adclient auditing messages. You can separately enable syslog facilities for logging general adclient messages, adclient auditing messages, and adnisd messages.

Select a value for this group policy from the list box, which contains a list of valid syslog facilities, for example, auth, authpriv, daemon, security, user, local *n*, and so on. The available facilities may vary depending on the operating system. The default value is auth.

If this group policy is not enabled, the audit messages are logged in the facility defined for the Set general audit logging facility policy.

This group policy modifies the logger.facility.adclient setting in the agent configuration file.

Rather than using the policy to set the facility, you can edit the agent configuration file to set the logger.facility.adclient parameter to any valid syslog facility. For example, you can set this parameter to log messages to one of auth, authpriv, daemon, security, local *n* facilities, and so on.

Set General Audit Logging Facility

Specify the syslog facility to use for logging general adclient activity. You can separately enable syslog facilities for logging general adclient messages, adclient auditing messages, and adnisd messages.

Select a value for this group policy from the list box, which contains a list of valid syslog facilities, for example, auth, authpriv, daemon, security, user, local *n*, and so on. The available facilities may vary depending on the operating system. The default value is auth.

This group policy modifies the logger.facility.* setting in the agent configuration file.

Rather than using the policy to set the facility, you can edit the agent configuration file to set the logger.facility parameter to any valid syslog facility. For example, you can set this parameter to log messages to one of auth, authpriv, daemon, security, local *n* facilities and so on.

You may also edit he agent configuration file to specify other process names for logging, or use an asterisk (*) to specify the default facility to use for all agent processes. For example, you can specify logger.facility.*: auth in the configuration file to direct all agent processes send messages to the auth facility of syslog.

Set Log Message Queue Size

This policy controls the maximum size in KB to use for queued log messages. The messages in the queue are sent to syslog asynchronously. During normal operation, if the size of the message queue reaches the value set for this parameter, no new messages are added until the size of the queue decreases below the maximum size you have specified. If the logging level is set to DEBUG, however, this policy's value is automatically multiplied by a factor of 4 to allow additional messages to be logged.

The value must be a positive integer. For example: 256

Setting this parameter to zero (0) disables the message queue, and causes all log messages to be written to the syslog facility synchronously. In most cases, disabling the message queue degrades system performance, and in extreme cases, may cause a dead lock with the syslog daemon during log rotations. Therefore, Delinea recommends that you never set this parameter value to 0.

This group policy modifies the log.queue.size setting in the agent configuration file. If this parameter is not defined in the configuration file, its default value is 256 KB.

Set NIS Audit Logging Facility

Specify the syslog facility to use for logging adnisd operations.

You can separately enable syslog facilities for logging general adclient messages, adclient auditing messages, and adnisd messages.

Select a value for this group policy from the list box, which contains a list of valid syslog facilities, for example, auth, authpriv, daemon, security, user, local *n*, and so on. The available facilities may vary depending on the operating system. The default value is auth.

If this group policy is not enabled, the audit messages are logged in the facility defined for the Set general audit logging facility policy.

This group policy modifies the logger.facility.adnisd setting in the agent configuration file.

Rather than using the policy to set the facility, you can edit the agent configuration file to set the logger.facility.adnisd parameter to any valid syslog facility. For example, you can set this parameter to log messages to one of auth, authpriv, daemon, security, local *n* facilities, and so on.

Login Settings

Use the group policies under **Login Settings** to control the following login and local account configuration options:

- Allow Localhost Users
- Allow Offline Login when User Account is Locked Out
- Enabled nss Emergency Shell
- Manage Login Filters
- Set Minimum Group ID (Lookup)
- Set Minimum User ID (Lookup)
- Set Sync Mapped Users
- Specify Group Names to Ignore
- Specify the Certificate Files to Add (Lookup)
- Specify the Fingerprints of Certificate Files to Ignore (Lookup)
- Specify User Names to Ignore
- Split Large Group Membership

Allow Localhost Users

Specify user names that should be allowed to authenticate locally when logging in.

This group policy is used to ensure that an account mapped to an Active Directory user can still access a system locally if there are problems with the network, the Active Directory server, or the agent.

If you select **Enabled** for this group policy, the users you specify can log in locally by appending @localhost to the user name. For example, if you specify the root user, you would log in as root@localhost.

This group policy modifies the pam.allow.override setting in the agent configuration file.

> **Note**: This group policy and the pam.allow.override configuration parameter are not supported on AIX computers. There is no equivalent policy or parameter for controlling local access on AIX computers.
>
> **Note**: If you are using a Solaris machine with the Name Switch Cache Daemon (NSCD) running, you will not be able to log in as an override user using <*username*>@localhost.

Allow Offline Login when User Account is Locked Out

Use this group policy to specify whether to allow a user to log in to a machine that is in disconnected mode if their account is locked.

If this policy is set to **Disabled**, or **Not configured**, by default, users with locked accounts cannot access a disconnected machine.

This group policy modifies the secedit.system.access.lockout.allowofflinelogin parameter in the agent configuration file.

Enabled nss Emergency Shell

Use this group policy to specify whether to use the default login shell when a user or group attempting to access the computer is not allowed to log on.

The default no-login shell and its location is typically platform-specific. For example, on Red Hat Linux the default shell for users who are denied access is:

/sbin/nologin

If this policy is **Disabled** or **Not configured**, by default, the nologin shell specified in the agent configuration file by the configuration parameter, nss.shell.nologin, is returned.

This group policy modifies the nss.shell.emergency.enabled parameter in the agent configuration file.

Manage Login Filters

Specify the users and groups allowed to log in to the system. With this policy, you can explicitly list either:

- Users and groups who are allowed to log in (all other users and groups are denied)
- Users and groups who should be denied access (all others are allowed)

When you enable this policy, you can select either the **allow** or **deny** option, then specify a list of user names, a list of group names, or both.

You can specify a list of users or groups in either of these ways:

- Enter a comma-separated list of users, groups, or both in the appropriate text boxes.
- Click the **List** button, then **Add**, to browse for and select users or groups to allow or deny.

Depending on your selections when you configure this group policy setting, the policy can modify any of the following configuration parameters in the agent configuration file:

pam.allow.groups

pam.allow.users

pam.deny.groups

pam.deny.users

> **Note**: This group policy does not support one-way, cross-forest groups.

Set Minimum Group ID (Lookup)

Specify the lowest group ID that is looked up in Active Directory.

> **Note**: This group policy does not apply to agent versions 4.1 or later. If you are using 4.1 or later, use the Specify Group Names to Ignore group policy to explicitly identify user groups that are always treated as local.

This group policy modifies the nss.mingid setting in the agent configuration file.

Set Minimum User ID (Lookup)

Specify the lowest user ID that is looked up in Active Directory.

> **Note**: This group policy does not apply to agent versions 4.1 or later. If you are using 4.1 or later, use the Specify User Names to Ignore group policy to explicitly identify user names that are always treated as local.

This group policy modifies the nss.minuid setting in the agent configuration file.

Set Sync Mapped Users

Synchronize the Active Directory password for local mapped users. When you enable this policy for a mapped user, if the user changes their Linux, UNIX, or Mac OS X password with the passwd command, or with a similar command, PAM changes the password to match in the local Linux, UNIX, or Mac OS X account. In this way, if there are problems with the network, Active Directory, or adclient, local users can still log into the machine.

> **Note**: This policy has no effect on Mac OS X computers.

To log in as a local user, append @localhost to the username. For example, log on as:

```
root@localhost
```

After enabling this policy, click **Browse** to search for users to add.

For this policy to work:

- The specified user must be a mapped user configured in centrifydc.conf with the pam.mapuser parameter.
- Either the Centrify or Microsoft password synchronization service must be installed on all domain controllers.
- The zone to which the machine belongs must be configured to support agentless clients.
- The Active Directory user to whom the local user is mapped must have a profile in the zone configured for agentless authentication.

This group policy modifies the pam.sync.mapuser setting in the agent configuration file.

Specify Group Names to Ignore

You can enter the list of local group names that aren't stored in Active Directory and separate each name with a space. The service will then use this list to disable looking up Active Directory account information for the specified groups. Ignoring this list of groups results in faster name lookups for system user accounts, such as tty and disk.

You can also specify a file that lists the usernames by entering the file: keyword and a file location. For example:

```
file:/etc/centrifydc/group.ignore
```

When you enable this policy, you can select the location where the group name list is populated. The default setting is "Populate group names to centrifydc.conf".

If you select "Populate group names to centrifydc.conf", this group policy modifies the nss.group.ignore setting in the DirectControl configuration (centrifydc.conf).

If you select "Populate group names to group.ignore", this group policy modifies the nss.group.ignore setting in centrifydc.conf as "file:/etc/centrifydc/group.ignore", and populates all configured group names to the group.ignore file. If you enter the file: keyword and a file location instead of the list of group names, this policy restores the ignore file /etc/centrifydc/group.ignore with the local list.

> **Note**: The selection of the populating location was added after DirectControl Agent version 5.6. If you're using version 5.5 or earlier, the agent ignores the population location setting and populates the user names to centrifydc.conf.

Specify the Certificate Files to Add (Lookup)

Define a list of certificate files which will be included in the certgp.pl install, if found.

It can be a list of certificates to be added. For example:

gp.mappers.certgp.pl.additional.cafiles: <ca-file> <ca-file> ...

It can also point to a file that contains a list of certificate files to be added. For example:

gp.mappers.certgp.pl.additional.cafiles: file:/etc/centrifydc/cert_included.list

The default value is empty.

This group policy modifies the gp.mappers.certgp.pl.additional.cafiles setting in the agent configuration file.

Specify the Fingerprints of Certificate Files to Ignore (Lookup)

Define a certificate list which will be excluded from the certgp.pl install, if matched.

It can be a list of fingerprints of the certificates to be excluded. For example:

gp.mappers.certgp.pl.exclude.cacerts: \<fingerprint\> \<fingerprint\> ...

It can also point to a file that contains a list of fingerprints of the certificates to be excluded. For example:

gp.mappers.certgp.pl.exclude.cacerts: file:/etc/centrifydc/cert_excluded.list

The default value is empty.

Specify User Names to Ignore

You can enter the list of local user names that aren't stored in Active Directory and separate each name with a space. The service will then use this list to disable looking up Active Directory account information for the specified users. Ignoring this list of users results in faster name lookups for system user accounts, such as tty and disk.

You can also specify a file that lists the usernames by entering the file: keyword and a file location. For example:

```
file:/etc/centrifydc/user.ignore
```

When you enable this policy, you can select the location where the user name list is populated. The default setting is "Populate user names to centrifydc.conf".

If you select "Populate user names to centrifydc.conf", this group policy modifies the nss.user.ignore and pam.ignore.users settings in the DirectControl configuration file (centrifydc.conf).

If you select "Populate user names to user.ignore", this group policy modifies the nss.user.ignore and pam.ignore.users settings in centrifydc.conf as "file:/etc/centrifydc/user.ignore", and populates all configured user names to the user.ignore file. If you enter the file: keyword and a file location instead of the list of user names, this group policy restores the ignore file /etc/centrifydc/user.ignore with the local list.

> **Note**: The selection of the populating location was added after DirectControl Agent version 5.6. If you're using version 5.5 or earlier, the agent ignores the population location setting and populates the user names to centrifydc.conf.

Split Large Group Membership

Specify whether you want to split up or truncate large groups. In operating environments that don't support large groups, commands that return group information may fail or return incomplete results when a group has a membership list that exceeds the maximum size allowed. Typically, the maximum size allowed for groups is 1024 bytes, which is roughly equivalent to 125 users. If you have large groups that exceed the 1024-byte limit, you can set this parameter to true to have those groups automatically split into multiple groups when they reach the maximum size.

The default value is true for Solaris, HPUX, and IRIX but false for all other operating environments.

> **Note**: This policy has no effect in Mac OS X environments.

This group policy modifies the nss.split.group.membership setting in the agent configuration file.

MFA Settings

Use the group policies under **MFA Settings** to control the following multi-factor authentication configuration options.

Enable Multi-Factor Authentication for Auto Zone and Classic Zone

Specify whether multi-factor authentication is **Enabled** for a classic zone or an Auto Zone. If you enable this policy, you can specify which Active Directory users and groups require multi-factor authentication to log on to their computers or to use privileged commands using the following group policies:

- Specify AD Users that Require Multi-Factor Authentication
- Specify AD Groups that Require Multi-Factor Authentication

This policy does not affect multi-factor authentication settings in hierarchical zones.

Before enabling this policy, you should be aware that multi-factor authentication relies on the infrastructure provided by the Delinea Platform.

Muti-factor authentication is disabled by default.

This group policy modifies the adclient.legacyzone.mfa.enabled configuration parameter in the agent configuration file.

Note that on computers running Centrify Express agents, you must set this policy using the configuration parameter. Group policies are not supported for Express agents.

Set Background Fetch Interval for Groups that Require Multi-Factor Authentication

Use this group policy to specify how often the Centrify Agent updates the cache with the list of users and groups in classic zones and Auto Zones that require multi-factor authentication, as well as the list of rescue users.

This is a background process that updates the cache periodically according to the interval specified (in minutes).

To disable this process, set the interval value to 0.

The default policy value is 30 minutes.

> **Note**: On computers running Centrify Express agents, you must set this policy using the configuration parameter. Group policies are not supported for Express agents.

This group policy modifies the adclient.legacyzone.mfa.background.fetch.interval configuration parameter in the agent configuration file.

Specify Centrify Identity Platform Tenant ID for Multi-Factor Authentication

Use this policy to specify the Delinea Platform tenant ID for multi-factor authentication.

This policy applies to Auto Zones and classic zones only.

You can get the Delinea Platform tenant ID from your service registration.

This policy modifies the adclient.legacyzone.mfa.tenantid setting in the agent configuration file.

Specify AD Users that can Login when Multi-Factor Authentication is Unavailable

Use this policy to specify rescue users who can log on to computers in a classic zone or an Auto Zone when multi-factor authentication is required, but the agent cannot connect to the Delinea cloud service.

You should specify at least one user account for this policy to ensure that someone can access the computers in the event that multi-factor authentication is unavailable.

If you enable this policy, you can specify users by name in the following formats:

- SAM account name: sAMAccountName
- SAM account name of a user in a different domain: sAMAccountName@domain
- User Principal Name: name@domain
- Canonical Name: domain/container/cn
- Full DN: CN=commonName,...,DC_domain_component,
- DCdomain_component
- An asterisk (*), which includes all Active Directory users

By default, this policy does not specify any rescue users.

This group policy modifies the adclient.legacyzone.mfa.rescue.users configuration parameter in the agent configuration file.

Specify AD Groups that Require Multi-Factor Authentication

Specify the Active Directory groups in classic zones or Auto Zones that are required to use multi-factor authentication to log on or use privileged commands.

For example, if you want to require all members of the Qualtrak Admin group to use multi-factor authentication when they log on to computers that host sensitive information, you can specify that group in this policy. Groups specified in this parameter must be security groups. Distribution groups are not supported.

If you enable this policy, you can specify groups by name in the following formats:

- sAMAccountName
- sAMAccountName@domain
- domain/container/cn

By default, no groups are required to authenticate using multi-factor authentication.

> **Note**: On computers running Centrify Express agents, you must set this policy using the configuration parameter. Group policies are not supported for Centrify Express agents. This group policy modifies the adclient.legacyzone.mfa.required.groups configuration parameter in the agent configuration file.

Specify AD Users that Require Multi-Factor Authentication

Specify the Active Directory users in classic zones or Auto Zones that require multi-factor authentication to log on or use privileged commands.

If you enable this policy, you can specify users by name in the following formats:

- sAMAccountName
- sAMAccountName@domain
- userPrincipalName@domain
- domain/container/cn
- CN=commonName,...,DC=domain_component,DC=domain_component
- An asterisk (*), which includes all Active Directory users

By default, no users are required to authenticate using multi-factor authentication.

> **Note**: On computers running Centrify Express agents, you must set this policy using the configuration parameter. Group policies are not supported for Express agents.

This group policy modifies the adclient.legacyzone.mfa.required.users configuration parameter in the agent configuration file.

Specify Delinea Identity Platform URL for Multi-Factor Authentication

Specify which Delinea Platform instance URL the agent will access in order to implement multi-factor authentication for users in classic zones and Auto Zones.

Enable this policy if you have access to more than one instance URL. If you have multiple instance URLs and do not specify which one the agent should use for multi-factor authentication, MFA will fail.

If you only have a single platform instance URL for all of the connectors in your Active Directory forest, the agent will use this URL for multi-factor authentication by default, and you do not need to enable this policy.

When specifying a cloud URL, the URL should be in the following format:

```
https://tenantid.domainfqdn:port/
```

For example:

```
https://abc0123.mydomain.com:443/
```

Note that on computers running Centrify Express agents, you must set this policy using the configuration parameter. Group policies are not supported for Express agents.

This group policy modifies the `adclient.legacyzone.mfa.cloudurl` configuration parameter in the agent configuration file.

Network and Cache Settings

Use the group policies under **Network and Cache Settings** to control connection timeout and object expiration intervals.

Blacklist DNS DC Hostnames

Specify a list of domain controllers to filter out when resolving the domain controller for the agent to contact through DNS. Set this policy to prevent the agent from attempting to contact a domain controller that you know is inaccessible, for example, because it resides behind a firewall, or from contacting a domain controller that is inappropriate because of its physical location, or because it is no longer a valid domain controller for the site.

To specify a domain controller, select **Enabled**, then click **Add** and enter the fully qualified name of a domain controller. For example:

wink2-admin13@ajax.com

You may enter only one controller at a time. To remove a controller from the list, select it and click **Remove**.

This group policy modifies the dns.block setting in the agent configuration file.

Enable LDAP Cross-Forest Search

Specify whether to allow the Centrify Agent to query trusted domains and forests for transitive trust information. If you enable this policy by selecting the **LDAP Cross-Forest Search** box, the agent generates a krb5.conf that includes information from all trusted forests and can be used to authenticate cross-forest users to Kerberos applications. If you disable this policy, the agent does not query external trusted domains or forests for information.

By default, the configuration parameter set by this policy is **Enabled**.

Querying external trusted forests can take a significant amount of time if the other forests are blocked by firewalls. You may want to set this parameter to false if your trust relationships, network topology, or firewalls are not configured properly for access.

This group policy modifies the adclient.ldap.trust.enabled setting in the agent configuration file.

Enable User Lookup and Login by CN

Specify whether you want to allow users to be found by their common name (cn) attribute.

By default, Centrify allows users to login using their Linux, UNIX, or Mac OS X profile name. In addition, Linux and Unix users can use their Active Directory displayName or Active Directory cn attribute (default value is disabled for Mac OS X users). Allowing users to log on using these additional attributes can require the agent to perform multiple searches to locate a user account in Active Directory. In environments with domain controllers under heavy load or with large user populations, searching Active Directory multiple times may negatively impact performance.

If you want to prevent the Centrify Agent from attempting to access to user information by the common name, you can disable this policy.

This group policy modifies the adclient.user.lookup.cn setting in the agent configuration file.

Enable User Lookup and Login by displayName

Specify whether you want to allow users to be found by their display name (displayName) attribute.

By default, Centrify allows users to login using their Linux, UNIX, or Mac OS X profile name. In addition, Linux and Unix users can use their Active Directory displayName or Active Directory cn attribute (default value is disabled for Mac OS X users). Allowing users to log on using these additional attributes can require the agent to perform multiple searches to locate a user account in Active Directory. In environments with domain controllers under heavy load or with large user populations, searching Active Directory multiple times may negatively impact performance.

If you want to prevent the Centrify Agent from attempting to access to user information by the display name, you can disable this policy.

This group policy modifies the adclient.user.lookup.display setting in the agent configuration file.

Force DNS to Use TCP

Force all DNS requests to use TCP rather than UDP. The initial size of the buffer is determined by the Set DNS UDP buffer size group policy (if you have enabled it), but the size will be increased, if necessary, for a specific response.

This group policy modifies the `dns.forcetcp` setting in the agent configuration file.

Force DNS to Rotate

Force all DNS queries to rotate through the list of servers in the /etc/resolv.conf file.

This group policy modifies the dns.rotate setting in the agent configuration file.

Force Switching to Different Domain Controller in the Preferred Site Periodically

This group policy specifies whether to force LDAP binding to be refreshed even if the current binding is to a local (preferred) Active Directory site. Under some conditions, binding to a different site can help facilitate load balancing between servers.

If you set this policy to **Enabled**, the agent will attempt to connect to another local domain controller when the period specified in the configuration parameter, adclient.binding.refresh.interval expires.

If this policy is set to **Disabled** or **Not configured**, by default, the agent will not attempt to connect to another domain controller if it is already connected to a preferred Active Directory site.

This group policy modifies the adclient.binding.refresh.force parameter in the agent configuration file.

Set Cache Negative Life Time

Specify the maximum time, in minutes, a negative object should remain in the cache. A negative object is returned when an object is not found in a search result. This policy determines how long that negative result should remain in the cache, regardless of the object type or object expiration time. By storing this negative result in the cache, the agent does not need to connect to Active Directory to look for an object that was previously not found.

The default period of time for keeping negative results is 5 minutes. Setting the policy value to 0 keeps negative objects in the cache indefinitely.

This group policy modifies the adclient.cache.negative.lifetime setting in the agent configuration file.

Set DNS Cache Size

Use this group policy with agent versions earlier than 4.5. This feature was deprecated starting with agent version 4.5.

Specify the unique number of DNS requests that can be cached by adclient. Set this value to approximately 10 times the number of unique domains in the forest.

This group policy modifies the adclient.dns.cache.size setting in the agent configuration file.

Set DNS Cache Timeout

Use this group policy with agent versions 4.5 and later. With agent versions earlier than 4.5, use the **Set DNS cache timeout (deprecated)** group policy.

Specify the maximum time, in seconds, before a cached DNS response expires. The default value is 300 seconds.

This group policy modifies the dns.cache.timeout setting in the agent configuration file.

Set DNS Cache Timeout (Deprecated)

Use this group policy with agent versions earlier than 4.5. This feature was deprecated starting with agent version 4.5. With agent versions 4.5 and later, use the **Set DNS cache timeout** group policy.

Specify the maximum time, in seconds, before a cached DNS response expires. The default value is 300 seconds.

This group policy modifies the adclient.dns.cache.timeout setting in the agent configuration file.

Set DNS UDP Buffer Size

Specify the maximum size of a UDP request in bytes. If the response is larger than this size, switch to TCP. If you have set the Force DNS to use TCP policy (dns.forcetcp parameter), the value you set here for the UDP buffer is the initial size of the TCP request buffer; the size will automatically be increased, if necessary, for a specific response.

The default value is 4096; the minimum is 512.

This group policy modifies the dns.max.udp.packet setting in the agent configuration file.

Set Domain DNS Refresh Interval

Use this group policy with agent versions earlier than 4.5. This feature was deprecated starting with agent version 4.5.

Specify the number of minutes between DNS updates. Specify a positive integer. The default value is 15 minutes.

This group policy modifies the `adclient.dns.update.interval` setting in the agent configuration file.

Set GC Expiration

Specify the maximum time, in seconds, that Distinguished Names are kept in the global catalog cache.

This group policy modifies the adclient.cache.expires.gc setting in the centrifydc.conf configuration file. By default, this parameter is set to 3600 seconds (1 hour).

Set Group Object Expiration

Specify the maximum time, in seconds, that a group object is kept in the local cache.

This group policy modifies the adclient.cache.expires.group setting in the agent configuration file. By default, this parameter is not defined in the configuration file, in which case, the value is determined by the Set Object Expiration group policy. If Set object expiration is not enabled, the default value is 3600 seconds (1 hour).

Set Idle Client Timeout

Specify the maximum time, in seconds, to wait before the agent closes a connection to an inactive client.

> **Note**: You must restart adclient for this policy to take effect.

This group policy modifies the adclient.client.idle.timeout setting in the agent configuration file.

Set LDAP Connection Timeout

Specify the maximum time, in seconds, for the agent to wait for a connection to an LDAP server to be established.

This group policy modifies the adclient.ldap.socket.timeout setting in the agent configuration file.

Set LDAP Response Timeout

Specify the maximum time, in seconds, for the agent to wait for a response from an LDAP server.

This group policy modifies the adclient.ldap.timeout setting in the agent configuration file.

Set LDAP Search Timeout

Specify the maximum time, in seconds, that the Active Directory Client Service will wait for a search response from an LDAP server.

This group policy modifies the adclient.ldap.timeout.search setting in the agent configuration file.

Set LDAP Trust Timeout

Specify the maximum number of seconds to wait for responses from external forests and trusted domains when attempting to determine trust relationships. If your trusted domains and forests are widely distributed, have slow or unreliable network connections, or are protected by firewalls, you may want to increase the value for this parameter to allow time for the agent to collect information from external domains and forests. The default value, if you do not set this policy, is 5 seconds.

This group policy modifies the adclient.ldap.trust.timeout setting in the agent configuration file.

Set LRPC Response Timeout

Specify the maximum time, in seconds, for an LRPC client to wait for a response.

This group policy modifies the `lrpc.timeout` setting in the agent configuration file.

Set LRPC2 Receive Timeout

Specify the maximum time, in seconds, for the agent to wait to receive data coming from a client request.

The default value is 30 seconds.

This group policy modifies the adclient.lrpc2.receive.timeout setting in the agent configuration file.

Set LRPC2 Send Timeout

Specify the maximum time, in seconds, for the agent to wait for reply data to be sent in response to a client request.

This group policy modifies the `adclient.lrpc2.send.timeout` setting in the agent configuration file.

Set Maximum Server Connection Attempts

Specify the maximum number of servers per domain the agent should attempt to connect to before going into disconnected mode. This policy is used if the agent is unable to connect to it's primary domain controller to enable it to query DNS for a list of other domain controllers and try each server in the list up to the maximum number of servers you specify. For example, if you have a large number of replica domain controllers for a given domain, you may want to use this policy to limit the number of servers for the agent to try in order to limit network traffic and improve performance.

The value should be a positive integer or 0. Setting the value to 0 means that the agent attempts to connect to every server in the list until successful.

The default value is 0.

This policy is ignored if you have defined a master domain controller for the zone to which the computer belongs because the computer only connects to that domain controller.

This group policy modifies the adclient.server.try.max setting in the agent configuration file.

This setting is deprecated for versions of adclient from 4.4.3 to 5.0.x. It is available in version 5.1.0 and later.

Set Object Expiration

Specify the maximum time, in seconds, before an object in the local cache expires. This expiration period applies to any object for which you have not set an object-specific expiration time, except Set GC Expiration, which has its own default value.

This group policy modifies the adclient.cache.expires setting in the agent configuration file. The default is 3600 seconds (1 hour).

Set Refresh Interval for Access Control Cache

Specify the maximum number of minutes to keep information from the authorization store cached before it expires.

The authorization store is an Active Directory object that stores the rights, roles, and role assignments that the privilege elevation service uses to control access to dzdo privileged commands, dzsh restricted environments, and PAM-enabled applications. Because the agent handles connecting to and retrieving information from Active Directory, this configuration parameter controls how frequently adclient retrieves the privilege elevation service set of information from Active Directory if any such data has been modified in Active Directory.

If local account management is enabled, this group policy also specifies how often etc/group and etc/passwd are updated on UNIX and Linux computers, based on the local group and local user settings that you configure in Access Manager.

If this policy is not **Enabled**, the default is 30 minutes.

Starting with agent version 5.1.3, this group policy modifies the adclient.refresh.interval.dz setting in the agent configuration file.

> **Note**: Prior to agent version 5.1.3, this group policy modified the adclient.azman.refresh.interval setting. That setting was deprecated in version 5.1.3.

Set UDP Timeout

Specify the maximum number of seconds to allow to complete UDP binding. The agent will attempt to bind twice. If the first bind request is not complete within the period specified by this policy, the agent sends a second request with a timeout period that is double the setting of this policy. If both bind requests fail to complete within the allotted time, the agent sets its status to disconnected.

For example, if you set this policy to 10 seconds and the bind request is not complete within 10 seconds, the agent sends a second bind request and waits a maximum of 20 seconds for the bind to complete before assuming the computer is disconnected from the network or Active Directory is unavailable.

The default value for this policy is 15 seconds.

This group policy modifies the `adclient.udp.timeout` setting in the agent configuration file.

Set User Object Expiration

Specify the maximum time, in seconds, that a user object is kept in the local cache.

This group policy modifies the adclient.cache.expires.user setting in the agent configuration file. By default, this parameter is not defined in the configuration file, in which case, the value is determined by the Set Object Expiration group policy. If Set object expiration is not enabled, the default value is 3600 seconds (1 hour).

Specify AD to NTLM Domain Mappings

Use the Specify AD to NTLM domain mappings group policy to manually map Active Directory domain names to NTLM domains. This parameter is useful when you need to use NTLM authentication and:

- firewalls prevent Kerberos authentication
- firewall constraints prevent the automatic discovery of Active Directory to NTLM domain mapping

To set this group policy, select **Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings > Specify AD to NTLM domain mappings**.

Provide the following information for the group policy:

- One or more pairs with ActiveDirectory domain name and NTLM domain name.
- Optionally, provide a file with a list of AD to NTLM domain name pairs. Include the file location. Use separate lines for each pair in the file. For example:

```
AJAX.ORG:AJAX FIREFLY.COM:FIREFLY
HR1.FIREFLY.COM:HR1
```

After you defined the mapping of Active Directory domains to NTLM domains, you can specify the list of domains that use NTLM authentication instead of Kerberos authentication. Use either the group policy, **Specify NTLM authentication domains** or the configuration parameter, pam.ntlm.auth.domains.

Alternative to using this group policy, **Specify AD to NTLM domain mappings**, you can use the adclient.ntlm.domains configuration parameter.

Specify DNS DC Hostnames

Specify the domain controller host names if your DNS is not configured to use Active Directory. In most cases, you should not use this group policy in a production environment because Active Directory automatically updates DNS with fail-over and replica servers optimized for the Active Directory site configuration. This group policy is used primarily for configuring an evaluation environment when the DNS server is on a Linux, UNIX, or Mac OS X computer and can't provide the _ldap service records.

The domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify DC host names:

After enabling this group policy, click **Add**, then enter the following information:

Domain: The domain name, for example, acme.com.

DC hostnames separated by space: One or more hostnames in the domain, for example, qa1-winxp, admin-winxp

Click **OK** to add the specified hostnames.

You can click **Add** again to add hosts from a different domain.

When you are done, click **OK**.

Once you've added one or more hostnames, you can select an existing domain and click **Edit** or **Remove** to edit or remove the specified hosts.

This group policy modifies the dns.dc.*domain_name* setting in the agent configuration file.

Specify DNS GC Hostnames

Specify the domain controller used as the global catalog if your DNS is not configured to use Active Directory. In most cases, you should not use this group policy in a production environment because Active Directory automatically updates DNS with fail-over and replica servers optimized for the Active Directory site configuration. This group policy is used primarily for configuring an evaluation environment when the DNS server is on a Linux, UNIX, or Mac OS X computer and can't provide the _gc service records.

The domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify GC hostnames:

After enabling this group policy, click **Add**, then enter the following information:

Domain: The domain name, for example, acme.com.

GC hostnames separated by space: One or more hostnames in the domain, for example, qa1-winxp, admin-winxp

Click **OK** to add the specified hostnames.

You can click **Add** again to add hosts from a different domain.

When you are done, click **OK**.

Once you've added one or more hostnames, you can select an existing domain and click **Edit** or **Remove** to edit or remove the specified hosts.

This group policy modifies the dns.gc.*domain_name* setting in the agent configuration file.

Specify IP Port Range that adclient Should Use

You can use this policy to specify the low and high ends of the range of IP ports for adclient and adnisd to use. These parameters control the outbound connection port for both TCP and UDP connections.

By specifying an IP port range, you can then configure your firewall to allow traffic through that port range only.

The typical port number is between 1024 and 65535. Setting this parameter does require a restart of adclient.

NIS Daemon Settings

Use the group policies under **NIS daemon** to control the operation of the Delinea Network Information Service (adnisd) on the local host computer. The Delinea Network Information Service provides a mechanism for the agent to respond to NIS client requests from computers not managed by Centrify Agents.

Set Thread Number for NIS Daemon

Specify the number of threads that may run simultaneously for the Centrify Network Information Service (adnis) on the local computer.

After enabling the policy, type a number or use the arrow keys to select a value. You must specify an integer between 1 - 200 inclusive. If you type a value outside this range, it is automatically reset to a valid number when you click **OK** or **Apply**.

The default value is 4 threads.

This group policy modifies the nisd.threads setting in the agent configuration file.

Specify NIS Daemon Update Interval

Specify the interval, in seconds, that the adnisd daemon waits between connections to Active Directory. At each interval, the adnisd daemon connects to Active Directory, gets the latest NIS maps for the local computer's zone, and updates its local NIS map data store.

The value must be an integer equal to or greater than zero. If the value is zero, then the update interval is disabled and the local NIS map data store is not updated. For example, to set the interval for getting NIS maps to 1 hour:

3600

If this group policy is not enabled, the default interval is 30 minutes (1800 seconds).

This group policy modifies the nisd.update.interval setting in the agent configuration file.

Specify Allowed NIS Mapping Files for NIS Daemon

Specify the name of the NIS maps currently available for NIS service. When the adnisd daemon connects to Active Directory, it retrieves the list of NIS maps available for the local computer's zone, creates a local map data store, and updates this configuration parameter, if necessary, to indicate the maps retrieved. If any NIS client requests a map that is not in the list specified by this group policy, the daemon refuses the request.

Enter a list of valid NIS map names, separated by spaces. You must explicitly specify the base maps and the derived maps. For example, to make the netgroup maps available but no other maps, enable this group policy and specify the following maps:

netgroup netgroup.byhost netgroup.byuser

If this group policy is not defined, all NIS maps found in Active Directory are retrieved and available for service.

This group policy modifies the nisd.maps setting in the agent configuration file.

Specify Disallowed NIS Mapping Files for NIS Daemon

Specify the name of the NIS maps you want to prevent the NIS service from using in response to NIS clients. This group policy enables you to exclude specific maps rather than explicitly specifying the maps you want to make available. For example, if you have a large number of automount maps or other network information that you want to make available to NIS clients but do not want to use agentless authentication, you can use this parameter to exclude the passwd and group maps but respond to automount or netgroup requests.

Enter a list of valid NIS map names, separated by spaces. Note that this policy excludes the named map and all derived maps; for example:

```
group passwd
```

If you do not enable this group policy, all NIS maps found in Active Directory are retrieved and available for service. This group policy overrides the setting of the Specify allowed NIS mapping files for NIS daemon.

This group policy modifies the nisd.exclude.maps setting in the agent configuration file.

Specify Allowed Client Machines for NIS Daemon

Specify a list of one or more subnets from which the daemon will accept NIS requests. You enable this group policy to restrict access to the Centrify Network Information Service by IP address. NIS requests that do not come from the IP addresses specified in this group policy are refused by the adnisd daemon.

You do not need to specify the local IP address for this group policy. The Centrify Network Information Service will always accept local NIS client requests.

The value must include both the specific IP address or subnet and the subnet mask, separated by a forward slash. For example:

192.168.111.0/255.255.255.0

You can specify multiple IP addresses by separating each IP address-subnet mask pair with a comma or a space. For example:

192.68.11.0/255.255.255.0,192.147.10.0/255.255.255.0

If this group policy or the parameter it modifies is not defined in the configuration file, only local NIS client requests are accepted by the adnisd process. When you enable this group policy, the default value is 0/0 to allow all machines.

This group policy modifies the nisd.securenets setting in the agent configuration file.

Set Switch Delay Time for NIS Daemon

Specify how long, in seconds, to wait before loading maps from a backup domain controller when the connection to the primary domain controller is lost. If the Centrify Network Information Service is unable to connect to its primary Active Directory domain controller, it will respond to NIS client requests using information in the local cache until the switch to the backup domain controller is complete.

The value must be an integer equal to or greater than zero. If the value is zero, then the delay is disabled. For example, to set the delay period to 2 hours, enter:

7200

If group policy is not enabled, the default delay for switching to the backup domain controller is ten minutes (600 seconds).

This group policy modifies the nisd.server.switch.delay setting in the agent configuration file.

Set Maximum Number of Mapping Files Allowed for NIS Daemon

Specify the number of alternate sets of NIS maps to retain. A new set of NIS maps is normally created when adnisd switches to an alternate domain controller. Keeping these alternate sets of maps allows Centrify Network Information Service to more efficiently switch between domain controllers.

You must specify an integer value greater than zero. The default is 2 map sets.

This group policy modifies the nisd.maps.max setting in the agent configuration file.

Set Large Group Suffix for NIS Daemon

Specify the suffix string or character to use in group names when automatically splitting up a group with a large number of members.

Because group.bygid and group.byname NIS maps often contain membership lists that exceed the 1024 limit of NIS data that can be served to clients, the adnisd process automatically truncates the membership list when this limit is reached. When you enable this group policy, the Centrify Network Information Service automatically splits a large group into as many new groups as needed to deliver the complete membership list.

When a group's data size exceeds the 1024 data limit, a new group is created. The new group name is formed using the original group name, followed by the string defined for this policy, and ending in a number that represents the numeric order of the new group created.

For example, for a large group named performix-worldwide-corp, a suffix string defined as -all, and the maximum length for group names as 10, the performix-worldwide-corp group membership is split into these multiple groups:

performix-worldwide-corp-all1 performix-worldwide-corp-all2 performix-worldwide-corp-all3 performix-worldwide-corp-all4

All of the new groups have the same group identifier (GID) as the original group. If the new group names would exceed the maximum length for group names on a platform, you can use the Set large group name length for NIS daemon group policy to set the maximum length for the new groups created.

If this policy is not enabled, the adnisd process truncates the group membership list such that each group entry is under 1024 characters.

This group policy modifies the nisd.largegroup.suffix setting in the agent configuration file.

Set Large Group Name Length for NIS Daemon

Specify the maximum number of characters to use in group names when groups with a large number of members are split into multiple new groups. Because some devices that submit NIS requests have limitations on the length of group names, you can use this parameter to specify the maximum length for group names.

When the adnisd process splits the group membership for a large group into multiple smaller groups, it truncates the original group name as needed to append the suffix defined in the Set large group suffix for NIS daemon group policy and not exceed the number of characters specified by this group policy. For example, if you have a large group named worldwide-all-corp, and have defined the suffix string as -all and the maximum length for group names as 10, when the worldwide-all-corp group membership is split into multiple groups, the groups are named as follows:

world-all1 world-all2 world-all3 world-all3

If this group policy is not enabled, the maximum group name length is 1024 characters by default.

This group policy modifies the nisd.largegroup.name.length setting in the agent configuration file.

Set Domain Name for NIS Daemon

Specify the NIS domain name for the adnisd process to use when communicating with NIS clients.

If you do not enable this group policy, the zone name is used by default.

This group policy modifies the nisd.domain.name setting in the agent configuration file.

Set Startup Delay Time for NIS Daemon

Specify the maximum time (in seconds) that adnisd will wait before answering NIS requests. If this policy is not enabled, adnisd begins answering requests only after all maps have been loaded or created, or when the default value, 180 seconds is reached, whichever comes first. If you set this policy, adnisd will begin answering NIS requests no later than the specified delay, as follows:

Before the delay time is reached, if all maps have not been loaded or created, requests are blocked waiting for the specified delay.

Once the delay time is reached, requests are answered whether all maps are loaded or not. Be aware that clients may receive partial or empty answers to their requests.

If all maps are loaded or created before the delay time is reached, adnisd will immediately begin answering requests.

Specify a value between 0 and 100000. If you enable the policy and do not change the value, the default is 180 seconds.

This group policy modifies the nisd.startup.delay setting in the agent configuration file.

NSS Overrides

Use the group policies under **NSS Overrides** to override entries in the local /etc/passwd or /etc/group files. These group policies provide additional access control and account configuration options on the computers where the policies are applied.

Specify NSS Group Overrides

Specify the group override entries you want to use in place of the entries in the local /etc/group file. You can use these settings to provide fine-grain control of the groups that can use the computer and to override the group ID for specific group accounts.

This group policy modifies the nss.group.override setting in the agent configuration file.

This group policy allows define filters to control the groups that can access a local computer. You can also use the override controls to modify the information for specific fields in each group entry on the local computer. For example, you can override the group ID or member list for a specific group on the local computer without modifying the group entry itself.

The syntax for overriding group entries is similar to the syntax used for overriding NIS. You use + and – entries to allow or deny access for specific groups on the local computer. Additional fields correspond to the standard /etc/group fields separated by colons (:).

> **Note**: If you don't specify override information for a field, the information from the local /etc/group file is used. You cannot specify override information for the password hash field, however. Any changes to this field in the override file are ignored and do not affect Centrify user passwords.

If you select **Enabled** for the **Specify NSS group overrides** group policy, you can type a comma-separated list of the override entries you want inserted into the override file, group.ovr, using the following format for each entry:

```
+zone_group_name:group_name:group_password:group_id:member_list

-zone_group_name:group_name:group_password:group_id:member_list
```

For example, you can specify entries similar to the following:

```
+users::::

+admins::::jdoe,bsmith,frank

+ftpusers:ftp::300:

-webusers

+::::
```

For more information about overriding group entries, see the sample group override file /etc/centrifydc/group.ovr.

Specify NSS Password Overrides

Specify the passwd override entries you want to use in place of the entries in the local /etc/passwd file. You can use these settings to provide fine-grain control of the users and groups who can use the computer and to override the user ID, group ID, default shell, or home directory for specific login accounts.

This group policy modifies the nss.passwd.override setting in the agent configuration file.

This group policy allows you to define filters to control access to a local computer. You can also use override controls to modify the information for specific fields in each /etc/passwd entry on the local computer. For example, you can override the user ID, primary group ID, default shell, or home directory for specific login accounts on the local computer without modifying the account entry itself.

The syntax for overriding passwd entries is similar to the syntax used for overriding NIS. You use + and – entries to allow or deny access for specific users on the local system. Additional fields correspond to the standard /etc/passwd fields separated by colons (:).

> **Note**: If you don't specify override information for a field, the information from the local /etc/passwd file is used. You cannot specify override information for the password hash field, however. Any changes to this field in the override file are ignored and do not affect Centrify user passwords.

If you select **Enabled** for the **Specify NSS password overrides** group policy, you can type a comma-separated list of the override entries you want inserted into the override file, passwd.ovr, using the following format for each entry:

```
+zone_username:username:password:uid:gid:GECOS:home_directory:shell

-zone_username:username:password:uid:gid:GECOS:home_directory:shell
```

For example, you can specify entries similar to the following:

```
+mike:::::::/usr/local/ultrabash

+jane@arcade.org:jdoe::300:300:::

+@sysadmins:::::::

-ftp

+@staff:::::::

+@rejected-users:::767:767:::/sbin/nologin
```

In the example above, the @ symbol denotes an Active Directory name. The name can be an Active Directory group name, a Centrify zone name, or some other container name. You can also specify an Active Directory user principal name (UPN) instead of the zone name.

Entries in the override file are evaluated in order from first to last with the first match taking precedence. This means the system will only use the first entry that matches a particular user. For example, if the user cruz is a member of both the staff group and the rejected-users group and you have defined the override entries as listed in the example above, the cruz user account is allowed to log on to the computer because the staff entry is evaluated and matched before the rejected-users entry. If the order were reversed in the override file, the cruz`` account would be flagged as a rejected-users account and denied access.

It is important, therefore, to consider the order in which you list the override entries in the group policy configuration. The order you use to specify the entries in the group policy is the order used when the entries are inserted into the override file.

Changes to the NSS password override entries only affect the entries inserted through the group policy. You can also manually create or update override entries in the override file on any local computer, if needed. Changes made to manually inserted or edited entries do not affect the entries maintained through the NSS Overrides group policies.

For more information about overriding passwd entries, see the sample password override file /etc/centrifydc/passwd.ovr.

PAM Settings

Use the group policies under **Pam Settings** to control a computer's PAM configuration.

Create Home Directory

Control whether a home directory should be created automatically when a new user logs on to a system for the first time.

This group policy should not be applied to computers that use NFS to mount home directories. By default, if this group policy is not configured, home directories are automatically created when new Active Directory users log on to a system for the first time except on Solaris computers.

If you do not want the Centrify Agent to automatically create user home directories, select **Disabled**. This group policy modifies the pam.homedir.create setting in the agent configuration file.

Create k5login

Create a .k5login file automatically in a user's home directory the first time the user logs on.

The .k5login file is used to enable Kerberos authentication and single sign-on in PAM-aware applications.

If you want Centrify Agent to automatically create the .k5login file in the user's home directory, select **Enabled**. This group policy modifies the pam.create.k5login setting in the agent configuration file.

Set Home Directory Permissions

Set the default read, write, and execute permissions on new home directories.

This group policy specifies the default permissions to assign a user's home directory if a new home directory is created for the user on the local computer.

If you want to set the permissions on the user's home directory, select **Enabled** then specify an octal value. For example, to give read, write, and execute permissions on the home directory to the user and no other permissions, type:

```
0700
```

This group policy modifies the `pam.homedir.perms` setting in the agent configuration file. The default value is `0755` on Mac OS X computers and `0700` on all other platforms.

Set Multi-Factor Authentication to Use an External PAM Module

This policy specifies the PAM application you want to use for multi-factor authentication if you are not using the Centrify PAM module and Privileged Access Service.

By default, the Centrify PAM module and Privileged Access Service are used to provide multi-factor authentication. This group policy allows you to specify the name of another PAM module if you would prefer to use a different multi-factor authentication provider.

This group policy modifies the pam.mfa.moduile.name setting in the agent configuration file.

Set Options for Multi-Factor Authentication by an External PAM Module

Specify the options to use if multi-factor authentication is done by an external PAM application.

> **Note**: Parameters must be separated by a space.

This group policy modifies the pam.mfa.module.options setting in the agent configuration file.

Set UID Conflict Message

Specify the message displayed if a user identifier (UID) conflict is detected during login. This message is displayed if there is a local user with the same UID but a different user name than the Active Directory user logging on.

When the message is displayed, the `%d` token in the message string is replaced with the UID of the conflicting local account. The message string you define must contain exactly one `%d` token, and no other string replacement (%) characters.

For example:

```
Account with conflicting UID (%d) exists locally
```

This group policy modifies the `pam.account.conflict.uid.mesg` setting in the agent configuration file.

For information about what to do when local conflicts are detected, see Set UID Conflict Resolution.

Set UID Conflict Resolution

Control how the Centrify Agent responds if a user logs on with an Active Directory account and either the Active Directory user name or Active Directory UID conflicts with a local user account.

The purpose of detecting a duplicate user name or duplicate UID is to prevent an Active Directory user from signing on and receiving privileges to modify files created by a different local user.

If you select **Enabled** for this group policy, you can choose one of the following options:

- ignore — Do not report duplicate user names or UID conflicts. If detected, log the conflict at the info level if logging is enabled.
- warn — Warn the user of the user name or UID conflict after a successful login. Log the conflict at warning level if logging is enabled. This is the default value.
- error — Report UID conflict to user after user name is entered. Don't accept password. Don't allow log in. Log conflict at error level.

This group policy modifies the pam.uid.conflict setting in the agent configuration file.

Set User Name and UID Conflict Message

Specify the message displayed if there are both user name and user ID conflicts detected during login. This message is displayed if there are two local account conflicts. For example, this message is displayed if there is a local user and the Active Directory user that have the same UID but different user names, and there is also another local account with the same user name as the Active Directory user but the two accounts have different UID values.

When the message is displayed, the `%s` token in the message string is replaced with the name of the first conflicting local account, and the `%d` token is replaced with the UID of the second conflicting local account. The message string you define must contain exactly one `%s` token and exactly one `%d` token, in that order, and no other string replacement (%) characters.

For example:

```
Accounts with conflicting name (%s) and UID (%d) exist locally
```

This group policy modifies the `pam.account.conflict.both.mesg` setting in the agent configuration file.

For information about what to do when local conflicts are detected, see Set UID Conflict Resolution.

Update Home Directory Ownership

Use this policy to specify whether or not to update the home directory ownership when the user logs in.

The default value is false.

> **Note**: You must set the pam.homedir.create parameter to true (the default value) for this policy to work.

Set User Name Conflict Message

Specify the message displayed if a user name conflict is detected during login. This message is displayed if there is a local user with the same user name but a different UID than the Active Directory user logging on.

When the message is displayed, the `%s` token in the message string is replaced with the name of the conflicting local account. The message string you define must contain exactly one `%s` token, and no other string replacement (%) characters.

For example:

```
Account with conflicting name (%s) exists locally
```

This group policy modifies the `pam.account.conflict.name.mesg` setting in the agent configuration file.

For information about what to do when local conflicts are detected, see Set UID Conflict Resolution.

Specify Message for Creating Home Directory

Specify the message to display when a user's home directory is created.

For example:

```
Creating home directory ...
```

This group policy modifies the pam.homedir.create.mesg setting in the agent configuration file.

Specify NTLM Authentication Domains

Use the Specify NTLM authentication domains group policy to specify the list of domains that use NTLM authentication instead of Kerberos authentication.

This group policy enables you to authenticate users behind a firewall when the Kerberos ports are blocked, but a trust relationship exists between domains inside and outside the firewall.

For example, use this group policy to specify that the Active Directory domains AJAX.ORG and FIREFLY.COM, which are outside of the firewall with a one-way trust to the forest inside the firewall, use NTLM authentication.

To set this group policy, select **Computer Configuration > Centrify Settings > DirectControl Settings > Pam Settings > Specify NTLM authentication domains**.

Provide the following information for the group policy:

- One or more fully-qualified Active Directory domain names.
- The Active Directory domain names that are mapped to NTLM domain names.

  These can be mapped automatically or manually:

  * automatically, if the firewall does not prevent the mapping from being discovered.
  * manually, if the firewall prevents the mapping from automatically being discovered, by modifying the contents of the `/etc/centrifydc/domains.conf` file.

  To manually configure the mapping use either the group policy, **Specify AD to NTLM domain mappings**, or the configuration parameter, adclient.ntlm.domains.

Alternative to using this group policy, **Specify NTLM authentication domains**, you can use the configuration parameter, pam.ntlm.auth.domains.

Specify Programs for which Multi-Factor Authentication is Ignored

Specify which PAM applications are exempt from multi-factor authentication.

For example, if you have a role with the login-all PAM application right and have selected the “Multi-factor authentication required” system right, you can use this group policy to bypass multi-factor authentication for programs that don't support it. You can also add program names to this list to skip multi-factor authentication when you wan to make specific exceptions to the MFA requirement.

By default, programs which are known to be unable to support multi-factor authentication are included in the list. For example, multi-factor authentication is ignored by default for the xscreensaver and vsftpd programs.

> **Note**: Program names must be separated by a space.

This group policy modifies the pam.mfa.program.ignore setting in the agent configuration file.

Password Prompts

Use the group policies under **Password Prompts** to customize the prompts displayed when Active Directory users are prompted to provide their password.

Set Account Disabled Error Message

Customize the text displayed during login if a user is denied access because the user's account is disabled. This group policy modifies the pam.account.disabled.mesg setting in the agent configuration file.

Set Account Expired Error Message

Customize the text displayed during login if a user is denied access because the user's account has expired.

This group policy modifies the pam.account.expired.mesg setting in the agent configuration file.

Set Account Locked Message for adpasswd

Customize the text displayed by the adpasswd program when users cannot change their password because their account is locked. This group policy modifies the adpasswd.account.disabled.mesg setting in the agent configuration file.

Set adclient Inaccessible Message

Customize the message displayed during password change, for a local Linux, UNIX, or Mac OS X user who is mapped to an Active Directory account, when the agent (adclient) is not accessible. This group policy modifies the pam.adclient.down.mesg setting in the agent configuration file.

Set Password Change Disallowed Message for adpasswd

Customize the text displayed by the adpasswd program when users are not allowed to change their password because password change for these users has been disabled in Active Directory. This group policy modifies the adpasswd.password.change.disabled.mesg setting in the agent configuration file.

Set Invalid User or Password Message for adpasswd

Customize the text displayed by the adpasswd program when a user enters an account name that is not recognized or an invalid password. This group policy modifies the adpasswd.account.invalid.mesg setting in the agent configuration file.

Set Permission Denied Message for adpasswd

Customize the text displayed by the adpasswd program when a user cannot change another user's password because of insufficient permissions. This group policy modifies the adpasswd.password.change.perm.mesg setting in the agent configuration file.

Set Lockout Error Message

Customize the text displayed when a user account is locked out. This group policy modifies the pam.account.locked.mesg setting in the agent configuration file.

Set Error Message for Empty Password Entered

Customize the text displayed when a user enters an empty password. Empty passwords are not allowed. This group policy modifies the pam.password.empty.mesg setting in the agent configuration file.

Set New Password's Mismatch Error Message for Password Change

Customize the text displayed during password change when the new passwords entered do not match. This group policy modifies the pam.password.new.mismatch.mesg setting in the agent configuration file.

Set Notification Text for Password Change

Customize the text displayed when Active Directory users attempt to change their password. This group policy modifies the pam.password.change.mesg setting in the agent configuration file.

Set Old Password Incorrect Error Message for Password Change

Customize the text displayed during password change when the old password entered is incorrect. This group policy modifies the pam.auth.failure.mesg setting in the agent configuration file.

Set Violation Error Message for Password Change

Customize the text displayed during password change if the operation fails because of a domain password policy violation. For example, if the user attempts to enter a password that doesn't contain the minimum number of characters or doesn't meet complexity requirements, this message is displayed. This group policy modifies the pam.policy.violation.mesg setting in the agent configuration file.

Set Password Prompt for Confirming New Password Change

Customize the text displayed when Active Directory users are prompted to confirm their new password. This group policy modifies the pam.password.confirm.mesg setting in the agent configuration file.

Set Password Prompt for New Password Change

Customize the text displayed when Active Directory users are prompted to provide their new password. This group policy modifies the pam.password.new.mesg setting in the agent configuration file.

Set Password Prompt for Old Password Change

Customize the text displayed when Active Directory users are prompted to provide their old password. This group policy modifies the pam.password.old.mesg setting in the agent configuration file.

Set Message Text for Password Change

Customize the text displayed when Active Directory users enter the correct password but must change the password immediately. This group policy modifies the pam.password.change.required.mesg setting in the agent configuration file.

Set Login Password Prompt

Customize the text displayed when Active Directory users attempts to log in. This group policy modifies the pam.password.enter.mesg setting in the agent configuration file.

Set Password Expiry Approaching Text

Customize the text displayed when the account password is approaching the expiration date. The message is displayed when the expiration date is within the limit defined by the pam.password.expiry.warn parameter. In the message, use the %d token for the number of days until expiration.

This group policy modifies the pam.password.expiry.warn.mesg setting in the agent configuration file.

Set Workstation Denied Error Message

Customize the text displayed during login if a user is denied access because of a workstation restriction. This group policy modifies the pam.workstation.denied.mesg setting in the agent configuration file.

Sudo Settings

Use the group policies under **Sudo Settings** to specify whether users must re-authenticate with sudo after logging out.

Force sudo Re-Authentication when Relogin

Specify whether users must authenticate again with sudo after logging out.

When a user authenticates with sudo, a ticket is temporarily created that allows sudo to run without re-authentication for a short period of time. If a user logs out and the ticket is not cleared, the ticket is reused when the user logs back in, and the user does not need to re-authenticate. If a user logs out and the ticket is cleared, the user must re-authenticate with sudo when logging back in.

Starting with release 2015, the way that you configure whether re-authentication is required depends on the tty_tickets parameter in the sudoers configuration file (/etc/sudoers.conf). In some situations, re-authentication requirements are also controlled by this policy. Details are as follows:

- If tty_tickets is enabled, tickets are always removed when a sudo user logs out, regardless of whether this policy is enabled or disabled. That is, when tty_tickets is enabled, this policy has no effect, and sudo users must always re-authenticate.
- If tty_tickets is disabled, the requirement for sudo users to reauthenticate is controlled by this policy and the adclient.sudo.clear.passwd.timestamp setting in the agent configuration file.

Tickets are cleared and sudo re-authentication is required in the following scenarios:

- The tty_ticket parameter in the sudoers configuration file is enabled (it is enabled by default)
- The tty_ticket parameter in the sudoers configuration file is disabled and this group policy is enabled
- The tty_ticket parameter in the sudoers configuration file is disabled and the adclient.sudo.clear.passwd.timestamp parameter is set to true

Tickets are not cleared and sudo re-authentication is not required in the following scenarios:

- The tty_ticket parameter in the sudoers configuration file is disabled and this group policy is disabled
- The tty_ticket parameter in the sudoers configuration file is disabled and the adclient.sudo.clear.passwd.timestamp parameter is set to false

By default, this policy clears tickets in the /var/run/sudo directory. To clear tickets in a different directory, use the adclient.sudo.timestampdir parameter in the agent configuration file as described in the *Configuration and Tuning Reference Guide*. This group policy modifies the adclient.sudo.clear.passwd.timestamp setting in the agent configuration file.

Window Settings

Use the group policies under **Centrify Settings > Windows Settings > Common Settings*** to control the configuration of Centrify Agents on Windows computers.

The following topics are covered:

Common Settings

Local Account Management

MFA Settings

Remote Authentication Dial-In User Service (RADIUS) Service Settings

Common Settings

Use the group policies under **Centrify Settings > Windows Settings > Common Settings** to control Centrify-managed Windows computers.

Configure Heartbeat Message for Centrify Analytics and SIEM (Windows)

Use this policy to specify how often (in minutes) Centrify Agent for Windows will send an information message to the Windows application log.

The Centrify Agent for Windows checks this setting every 5 minutes.

By default, this policy is set to zero (0), which means that this task is disabled.

Configure Windows Authentication Grace Period for Run with Alternate Account

You use this group policy to specify that there is a grace period for users running an alternate account before they must re-authenticate. By default, this policy is not enabled. If you enable this policy, you specify the time period in minutes. This policy works in conjunction with Require re-authentication to run application with alternate account.

You set up alternate accounts in Privileged Access Service. Alternate accounts are a way that you can allow a user to access a privileged account.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, there is no grace period for re-authentication for users running an application with an alternate account.
- When this policy is **Enabled**, you specify the grace period by the number of minutes. This grace period is how long the user can run an application using an alternate account before having to re-authenticate.

If you have not also enabled the Require re-authentication to run application with alternate account policy, this policy has no effect.

Configure Windows Authentication User Privilege Elevation Grace Period

You can use this group policy to configure the Windows authentication grace period (in minutes) for user privilege elevation, such as run as role, run with privilege, new desktop, and switch desktop.

This per-session grace period starts when the user performs a successful privilege escalation in the session and the grace period is restarted. If the group policy is set to:

- **Enabled**: the grace period for privilege elevation is configured in the group policy.
- **Disabled**: the grace period for privilege elevation is disabled.
- **Not Configured**: the grace period for privilege elevation is not enabled and a local policy can override the setting.

Custom Message for Locked User Accounts

Use the Custom message for locked user accounts policy to customize the message that will be shown to the user when the user tries to log into a locked user account.

- If this policy is set to **Enabled**, an administrator can specify the message that will be shown to the user when the user tries to log into a locked user account. The credential provider shows the message if the message is specified (not empty).
- If this policy is set to **Disabled** or **Not Configured**, you will see the same message as the windows credential provider.

The group policy "Custom message for locked user accounts" only changes the message for the console logon or remote logon without Network Level Authentication (NLA). If you log on remotely with NLA, Remote Desktop Client will block logon with its message.

Disable the Centrify Notification Icon

Disable the Centrify icon in the notification area of the Windows task bar for users that are not assigned any roles, or for machines that are not joined to a domain.

Enable Run with Alternate Account

You can use this group policy to enable the ability for users to run an application with an alternate account.

You set up alternate accounts in Privileged Access Service. Alternate accounts are a way that you can allow a user to access a privileged account.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, the user can run an application with only their user role if Centrify Privilege Elevation Service is also enabled.
  If only Centrify Identity Platform is enabled, then in order for a user to use an alternate account, they must log in to Privileged Access Service and check out the password directly.
- When this policy is **Enabled**, the user can run an application using an alternate account by right-clicking the application icon and selecting Run with Alternate Account.

Enable Setup Centrify Offline MFA Profile

Use this group policy to enable setup of Centrify offline MFA profile.

There are two settings for this group policy:

- When this policy is **Enabled** or **Not Configured**, you can set up the passcode for multi-factor authentication. A passcode can be used to fulfill multi-factor authentication in the event the computer cannot connect to the Centrify Identity Platform.
- When this policy is **Disabled**, you cannot setup an offline MFA profile.

Enable Use of Alternate User's Role to Run an Application

You can use this group policy to use an alternate user's role to run an application. The alternate user's credential is required when you use an alternate user's role. This policy does not apply to the `runasrole` command-line interface.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, the user can run an application with only their user role.
- When this policy is **Enabled**, the user can use another user's role to run an application.

Hide Command Line Arguments in Analytics

You can use this group policy to hide command line arguments from Analytics Data (RunWithPrivilege Events).

There are two settings for this group policy:

- By default, when this policy is **Enabled** or **Not Configured**, the Analytics Data does not show the command line arguments.
- If this policy is set to **Disabled**, the Analytics Data shows the command line arguments.

Prevent Local Administrators from Being Able to Log On in Rescue Mode (When There are No Explicit Rescue Users Defined)

Use this policy to prevent local administrators that are not defined rescue users from logging in to a machine that is running in rescue mode or Windows Safe Mode.

If you set this policy to **Enabled**, you should add users and groups to the rescue user list by issuing them the rescue user role, or a custom role with the rescue user system right selected.

If you are not joined to a zone (because your computers are not managed by Server Suite), you can enable the group policy, Specify a list of rescue users (when the agent is not joined to a zone), and add users to the rescue user list.

By default, if this policy is set to **Disabled** or **Not Configured**, all local administrators are able to log in without multi-factor authentication when the machine is running in rescue or safe mode.

Re-Authentication: Require Smart Card

Enable this policy to require Windows users to re-authenticate using a smart card.

By default, this setting is disabled.

Require Justification on Privilege Elevation

You can use this group policy to require any user to provide a reason when they operate with elevated privileges, such as run with privilege, run as role, and new desktop.

This group policy works in conjunction with the Specify a privilege elevation validator policy. If you only set one of these policies, any affected user is prompted to provide a reason for privilege escalation.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, users can run with elevated privileges as normal.
- When this policy is **Enabled**, the agent prompts the user with a justification dialog box, where the user can provide a reason category and a text string for the reason.

  Also, if you've configured your system to work with a ticketing system such as ServiceNow, you can use the Specify a privilege elevation validator group policy to validate the ticket number that the user enters.

  You can view the reason information that users enter in the audit trail event.

You can use this group policy with loopback mode, so that you can apply the policy based on the computer that a user logs into. For more details about loopback mode, see the Microsoft documentation, such as the following page:

https://support.microsoft.com/en-us/help/231287/loopback-processing-of-group-policy

Require Re-Authentication to Run Application with Alternate Account

You use this group policy to specify that users running an alternate account must re-authenticate. By default, this policy is false.

You set up alternate accounts in Privileged Access Service. Alternate accounts are a way that you can allow a user to access a privileged account.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, after the user selects the option to run an application with an alternate account, they will not be prompted to re-authenticate.
- When this policy is **Enabled**, the user who runs an application using an alternate account will need to re-authenticate. To specify how long before the user is prompted for re-authentication, you define that grace period in the Configure Windows authentication grace period for run with alternate account policy.

Specify a List of Blacklisted Domains

Enable this group policy to specify a list of domains that will be ignored by the Centrify Agent.

After enabling this policy, enter one or more domain names, separated by a comma, in the following format:

```
domain1.com,domain2.com,...,domainN.com
```

If the root domain of a trusted forest is specified in this list, all of its leaf domains will also be ignored.

By default, if this policy is set to **Not configured**, no domains are blacklisted.

Specify a List of Rescue Users (When the Agent is not Joined to a Zone)

If the agent is not joined to a zone (because your computers are not managed by Server Suite), use this policy to specify a list of users who can log in without using multi-factor authentication if the machine is running in rescue mode or Windows Safe Mode.

The user name can be specified in any of the following formats:

- sAMAccountName
- sAMAccountName@domain (if the account is not in the current domain).
- UserPrincipalName@domain
- An asterisk (*), which includes all Active Directory users.

You can enter the list of users separated by a comma, for example:

joe, janedoe, user1, user2@domain.com.

By default, if this policy is set to **Disabled** or **Not configured**, only local administrators can log on in rescue mode or safe mode. However, if you enable Prevent local administrators from being able to log on in rescue mode (when there are no explicit rescue users defined), and do not enable this policy, no one will be able to log in if the computer is running in these modes.

Specify a List of Whitelisted Domains

Enable this group policy to specify a list of domains that will be trusted and processed by the Centrify Agent. If you enable this policy, **only** these domains will be trusted.

After enabling this policy, enter one or more domain names, separated by a comma, in the following format:

```
domain1.com,domain2.com,...,domainN.com
```

You must specify the root domain on this list if you specify any of its leaf domains.

By default, if this policy is set to **Not configured**, all domains are trusted and processed by the Centrify Agent.

Specify Offline MFA Profile Desktop Notification Message

Use this group policy to specify conditions for the offline MFA profile desktop notification message.

There are two settings for this group policy:

- **Enabled** or **Not Configured**: The user-specified offline MFA profile desktop notification message appears.
- **Disabled**: The offline MFA profile desktop notification message does not appear.

Specify a Privilege Elevation Validator

You can use this computer configuration group policy to validate ticket information that a user enters when she provides a ticket number along with a privilege elevation reason. You can validate ticket information using a customized PowerShell script against a ticketing system, such as ServiceNow.

If you enable this policy, here are some important things to know:

- Centrify provides a sample script that you can use as a starting point for your own script. At the minimum, you need to enter your ServiceNow URL for the $url parameter.
- You can get the sample script from github: in the centrify-agent-windows repo, go to Samples > ITSM validation > servicenow.
- If the ticket ID is not validated successfully, the user's request for elevated privilege is rejected.
- The custom PowerShell script must be available and accessible on each Windows computer where the validation occurs. If you're not running the PowerShell script on a local computer, be sure to allow remote PowerShell access for the script.
- This group policy works in conjunction with the Require justification on privilege elevation policy. If you only set one of these policies, any affected user is prompted to provide a reason for privilege escalation.
- If the script cannot validate the ticket entry within the specified timeout duration, then the validation fails. By default, the timeout value is 2 minutes.

Please consult the group policy explain text for more details.

There are two settings for this group policy:

- By default, when this policy is **Disabled** or **Not Configured**, users can run with elevated privileges as normal.
- When this policy is **Enabled**, you specify the PowerShell script filename and users entries are validated against the third-party ticketing system before granting privileged access.

You can view the reason information that users enter in the audit trail event.

Specify Whether to Keep the Desktop Notification Permanently Visible

Use this group policy to specify whether to keep the desktop notification message visible at all times.

The desktop notification message shows what roles are currently being used by the user and may be helpful to remind those using privileged desktops which elevated privileges they are authorized to use for that desktop. Select **Enabled** to keep the message visible.

If this policy is **Disabled**, or **Not configured**, by default, the notification message will fade a few seconds after clicking the Centrify icon in the system notification tray.

Local Account Management

Use the group policies under **Centrify Settings > Windows Settings > Local Account Management** to control local Windows users and groups on Centrify-managed Windows computers.

Enable Local Account Management Feature

Use this policy to specify that Windows local account management is enabled for the Centrify Agent for Windows.

By default this policy is set to Not Configured, which in this case means that Windows local account management is not enabled.

Enforce Local Account Management Feature

Use this policy to specify that Windows local account management is enforced for the Centrify Agent for Windows. By enforcing local account management, if you remove a user from a zone or computer in Access Manager, the user is removed from all affected computers.

By default this policy is set to Not Configured, which in this case means that Windows local account management is not enforced.

Synchronization Interval

Use this policy to specify how often (in seconds) the service synchronizes local users and groups information with affected computers.

By default this policy is set to 3600 seconds, which is the equivalent to 1 hour.

Notification Command Line

Use this policy to specify the command line script that the Centrify Agent for Windows runs after provisioning local users and groups. This policy only applies if the Windows local account management feature is enabled.

MFA Settings

Use the group policies under **Centrify Settings > Windows Settings > Centrify MFA Settings** to control multi-factor authentication on Centrify-managed Windows computers.

Configure Multi-Factor Authentication for Logon when the Agent Cannot Connect to the Platform

You can use this group policy to configure offline multi-factor authentication for users that are required to use multi-factor authentication to log on to their computers in the event that the agent cannot connect to the Centrify Identity Platform.

There are three configuration possibilities:

- If an offline MFA profile is setup, prompt for offline MFA. Otherwise don't allow to proceed.
- If an offline MFA profile is set up, prompt for offline MFA. Otherwise, allow to proceed and remind user to set up the offline MFA profile.
- Allow to proceed. Don't prompt for offline MFA.

If this policy is set to **Disabled** or **Not Configured**, the default is the second option.

Configure Multi-Factor Authentication for Privilege Elevation when the Agent Cannot Connect to the Platform

You can use this group policy to configure offline multi-factor authentication for users that are required to use multi-factor authentication to use elevated roles in the event that the agent cannot connect to the Centrify Platform.

There are three configuration possibilities:

- Only users who have set up an offline MFA profile will be prompted for offline multi-factor authentication for privilege elevation. Users who have not set up an offine passcode will not be able to proceed.
- If an offline MFA profile is set up, prompt for offline MFA. Otherwise, allow the user to proceed and remind them to set up the offline MFA profile.
- Users can use elevated rights or roles when their machine is offline without multi-factor authentication.

If this policy is set to **Disabled** or **Not Configured**, the default is the first option.

Connect to the Centrify Platform Directly

Connect to the Centrify identity platform directly without using a web proxy or a connector as a web proxy. If you enable this policy, you must configure the client to be able to connect to the identity service.

Continue with MFA Challenges after Failed Windows Authentication in Logon Screen

Configuring this policy setting allows you to continue with MFA challenges, even with a failed Windows authentication.

> **Note**: The following is recommended for PCI DSS or NIST 800-53 guidelines for multi-factor or multi-step authentication.

If this policy is set to **Enabled**, authentication on the Windows logon screen continues with MFA challenges with the wrong password or use of expired/locked out/disabled accounts.

> **Note**: Specify the multi-factor authentication grace period is disabled when this policy is enabled.

If this policy is set to **Disabled** or **Not Configured**, authentication on Windows logon screen fails immediately when you enter the wrong password and the MFA challenges are not triggered. To continue to the second MFA challenge when previous challenge response failed, use the policy "Continue with additional challenges after failed challenge" in the Admin Portal.

Disable Multi-Factor Authentication for Screen Unlock

Use this policy to disable multi-factor authentication for the Windows lock screen. When multi-factor authentication is required to log on to a Windows machine, by default, users must also use multi-factor authentication to unlock the Windows lock screen. If this policy is set to **Enabled**, users that require multi-factor authentication to log on will not have to use multi-factor authentication to unlock the Windows lock screen.

If this policy is set to **Disabled** or **Not configured**, the default is to require users to use multi-factor authentication to unlock the Windows lock screen.

Disable Self-Service Password Reset

You can use this group policy to allow the administrator to force disabling of the password reset feature. There are two settings for this group policy:

- **Enabled**: If this policy is set to **Enabled**, the self-service password reset feature on the machine is disabled, including the cloud-enabled self-service password reset.
- If this policy is set to **Disabled** or **Not Configured**, the self-service password reset feature on the machine follows the cloud policy setting (cloud policy settings can be found at: **Policy Settings > User Security Policies > Self Service > Password Reset**). The cloud policy settings are accessed through the Admin Portal.

  Note: The admin portal is available after you log in to a Centrify Platform instance.

Enable Multi-Factor Authentication for Windows Login (when the Agent is not Joined to a Zone)

Use this policy to enable multi-factor authentication for Windows login when the agent is not joined to a zone.

If this policy is set to **Disabled** or **Not configured**, the default is that no user is required to use multi-factor authentication to log in.

Force to Enter Explicit UPN

Configure this policy setting to force all users that require MFA to log in to the machine using the UPN format of: user@domain.com. There are two settings for this group policy:

- If this policy is set to **Enabled**, all users that require MFA must log in using the UPN format, otherwise an error message appears "Invalid User. Please use format user@domain.com and try again."

  Note: All users that do not require MFA can log in using either the UPN format or NT account format.

- If this policy is set to **Disabled** or **Not Configured**, all users can log in using either the UPN format or NT account format.

Send UUID for MFA Challenges

Configure this group policy to enable the DirectAuthorize Agent to send user UUID with the user UPN for the MFA challenges.

- If this policy is set to **Enabled**, DirectAuthorize Agent sends the user UUID as addition field with the user UPN for the MFA challenges.
- If this policy is **Disabled** or **Not Configured**, DirectAuthorize Agent sends only the UPN for the MFA Challenges.

Skip Client Certificate Authentication

Use this group policy to skip client certificate authentication to the Centrify Platform if client certificate authentication is disabled or blocked by enterprise policies or proxy settings.

If you enable this policy, you must configure the client to be able to connect directly to the Centrify Connector for multi-factor authentication.

If this policy is set to **Disabled** or **Not configured**, client certificate authentication is required for multi-factor authentication.

Specify a Web Proxy URL

Specify a web proxy to use to connect to the Centrify identity platform. If you have enabled the client to connect to the cloud service directly, without using a connector or web proxy, enabling this policy has no effect.

Specify Active Directory Users that Require Multi-Factor Authentication on Windows Login (when the Agent is not Joined to a Zone)

Use this policy to specify the Active Directory users that are required to use multi-factor authentication to log on to Windows computers. If you enable this policy, you can specify users by name in the following formats:

- sAMAccountName
- sAMAccountName@domain
- userPrincipalName@domain
- An asterisk (*), which includes all Active Directory users

Use quotes for names containing spaces, for example, “Krusty T. Clown”.

By default, no users are required to authenticate using multi-factor authentication.

Specify How Frequently to Check for Responses to Multi-Factor Authentication Challenges

Set the polling interval in seconds for checking whether a user has responded to a multi-factor authentication challenge. Some authentication challenges require the client to wait for the user to respond to the challenge.

This value defines how frequently the client checks with the cloud service for a user's challenge response. The lower the value, the faster the client responds.

The minimum value you can specify is 1 second and the maximum value is 300 seconds. If you enable this policy, the default value is 3 seconds.

Specify the Multi-Factor Authentication Grace Period

You can use the following two group policies under **Windows Settings > MFA Settings** to control the multi-factor authentication grace period:

- Configure multi-factor authentication lock screen grace period
- Configure multi-factor authentication user privilege elevation grace period

The *Configure multi-factor authentication lock screen grace period* group policy allows the administrator to configure the multi-factor authentication grace period (in minutes) for the lock screen. If the group policy is set to:

- **Enabled**: the grace period for lock screen is enabled and it is configured in the group policy. If this value is configured to 0, it means no grace period for MFA in the lock screen.
- **Disabled**: the grace period for lock screen is disabled.
- **Not Configured**: the grace period for lock screen is not enabled and a local policy can override the setting.

The *Configure multi-factor authentication user privilege elevation grace period* group policy allows the administrator to configure the multi-factor authentication grace period for user privilege elevation, such as run with privilege and add new desktop. This per-session grace period starts when the user performs a successful MFA challenge in the session and the grace period is restarted. If the group policy is set to:

- **Enabled**: the grace period for privilege elevation is configured in the group policy.
- **Disabled**: the grace period for privilege elevation is disabled.
- **Not Configured**: the grace period for privilege elevation is not enabled and a local policy can override the setting.

Applying the MFA Lock Screen Grace Period to Remote Sessions

By default, the *Configure multi-factor authentication lock screen grace period* group policy applies only to console sessions and not to remote sessions. However, you can configure a registry key to apply the grace period to remote sessions too. You can deploy this registry key as an additional policy.

To enable the MFA lock screen grace period for remote sessions

- Add the following registry entry on each computer where you have installed the Centrify Agent for Windows:

  HKEY_LOCAL_MACHINE\SOFTWARE\Centrify\DirectAuthorize\Agent\ApplyLockScreenGracePeriodToRDPSessions = 1 (REG_DWORD)

Specify the Authentication Source for Privilege Elevation

Use this policy to specify the authentication source for privilege elevation. You can choose either multi-factor or RADIUS authentication.

If this policy is set to Enabled, agents will use the configured source for privilege elevation multi-factor authentication.

If this policy is set to Disabled, you cannot use another authentication source for privilege elevation multi-factor authentication.

If this policy is set to Not Configured, you can configure another authentication source locally on the agent.

Specify the Centrify Connector URL to Use

Specify the connector to use.

You should specify the URL with a fully-qualified domain name and port number. For example, if using a secure HTTP (HTTPS) connection, type an entry similar to the following:

```
https://acme.example.com:8080/
```

If you enable and apply this policy, you must also enable and apply the policies to specify the cloud instance URL and, if applicable, the web proxy URL.

If you don't configure this policy, the cloud instance URL will automatically locate an available connector to use by default.

Specify the Connection Timeout for Multi-Factor Authentication Requests

Use this group policy to set the connection timeout for multi-factor authentication requests. This policy defines the number of seconds to wait before the request times out.

If you enable this policy, the minimum value you can specify is 1 second, and the maximum value is 100 seconds.

If this setting is set to **Not Configured**, the default value is 15 seconds.

Specify Credential Providers to Exclude from the Logon Screen

Use this group policy to list specified credential providers to exclude from the login options on the Windows login screen when users access the machine remotely.

You must list the Class IDentifiers (CLSID) for the providers you would like to exclude. For example, to exclude the Windows Password Provider and the Smartcard Credential Provider on machines running Windows 8 or later and Windows Server 2012 or later, you would enter the following:

{60b78e88-ead8-445c-9cfd-0b87f74ea6cd},{8FD7E19C-3BF7-489B-A72C-846AB3678C96}

To find the CLISDs for installed credential providers, navigate to the following location in the HKEY_LOCAL_MACHINE registry:

SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers

If this policy is set to **Disabled** or **Not configured**, only the Windows Password credential provider will be disabled by default.

Specify the Platform Instance Id to Use (when the Agent is Not Joined to a Zone)

Use this policy to specify the Centrify Identity Platform instance Id also called a tenant ID) to use when the agent is not joined to a zone.

In most cases, this policy is only required if you have access to multiple platform Ids and want to explicitly specify which platform instance to connect. For example: AAH0305

You can get the Centrify Identity Platform tenant ID from your service registration.

If you're using a version of Access Manager prior to 19.6 and you upgrade your connectors to a version of 19.5 or later, please make sure you manually update the tenant URL to use the .net domain extension after the connector upgrade. Otherwise, MFA will not work and the Centrify Identity Services Platform won't be listed in the agent configuration.

If you're already using a version of Access Manager of 19.6 or later, you can set the tenant ID on the zone. Then when you upgrade your connectors, the agent gets the new tenant URL automatically.

Specify the Platform Instance URL to Use

Specify the Centrify Identity Service instance URL to use. In most cases, this policy is only required if you have access to multiple cloud instances and want to explicitly specify which instance to connect to.

You should specify the URL using the customer-specific identifier for the cloud instance and a fully-qualified domain name and port number.

For example, if using a secure HTTP (HTTPS) connection, type an entry similar to the following:

https:/ABC1234.my.centrify.net:443/

Specify the Platform Instance URL to Use (when the Agent is Not Joined to a Zone)

Use this group policy to specify which platform instance URL the agent will access for users of computers that are not joined to a zone.

Enable this policy if you have access to more than one instance URL.

If you only have a single authentication server URL for all of the connectors in your Active Directory forest, the agent will use this URL by default, and you do not need to enable this policy.

When specifying a URL, the URL should be in the following format:

https:/customerid.domainfqdn:port/

For example:

https:/abc0123.my.centrify.net:443/

Specify the Timeout on Skipping Previously Disconnected Centrify Connectors

Specify the length of time, in seconds, for the agent to ignore previously disconnected connectors while attempting to connect to the cloud for an authentication request.

You can avoid connection delays by specifying a longer timeout period for previously disconnected connectors. The agent will not attempt to connect with these connectors until the timeout period ends. The minimum value you can specify is 0 seconds and the maximum value is 86400 seconds. The default value is 1800 seconds.

Specify the timeout on Using the Last Successfully Connected Centrify Connector First

Specify the length of time, in seconds, for the agent to attempt to connect to the cloud using the last successful connector.

The lower you set this value, the faster the agent will try other connectors during the next authentication request. The minimum value you can specify is 0 seconds and the maximum value is 86400 seconds. The default value is 600 seconds.

Remote Authentication Dial-In User Service (RADIUS) Settings

Use the group policies under **Centrify Settings > Windows Settings > MFA Settings > Remote Authentication Dial-In Service (RADIUS) Settings** to control Radius multi-factor authentication on Centrify-managed Windows computers.

Enable Remote Authentication Dial-In User Service (RADIUS)

Use this policy to enable Remote Authentication Dial-In User Service (RADIUS) for privilege elevation.

If this policy is set to Enabled, you can use RADIUS for privilege elevation multi-factor authentication.

If this policy is set to Disabled, you cannot use RADIUS for privilege elevation authentication.

If this policy is set to Not Configured, you can enable RADIUS locally on the agent.

Specify the RADIUS Connection Timeout

Use this policy to specify the connection timeout in seconds for the RADIUS server.

If this policy is set to Enabled, agents will use the configured timeout for authentication.

If this policy is set to Disabled or Not Configured, you can configure the connection timeout locally on the agent.

Specify the RADIUS Server IP Address

Use this policy to specify the RADIUS server IP address.

If this policy is set to Enabled, agents will use the configured RADIUS server for authentication.

If this policy is set to Disabled or Not Configured, you can configure the RADIUS server IP address locally on the agent.

Specify the RADIUS Server Port Number

Use this policy to specify the RADIUS server port number.

If this policy is set to Enabled, the agent will use the configured port for authentication.

If this policy is set to Disabled or Not Configured, you can configure the RADIUS server port number locally on the agent.

Audit and Audit Trail Settings

This chapter describes the audit-related group policies that are located under **Centrify Audit Trail Settings** and **Centrify Audit Settings**.

Group policies located under **Centrify Audit Trail Settings** allow you to specify both category-specific and global audit trail targets. These group policies are located in subfolders under **Centrify Audit Trail Settings**. See Audit Trail Settings for details about these group policies.

Group policies located under **Centrify Audit Settings** allow you to configure the auditing agent installation, and platform-specific auditing features. These group policies are located in subfolders under **Centrify Audit Settings**. See Centrify Audit Settings for details about these group policies.

The following topics are covered:

Alternate Location for Policies Installed with an ADMX Template

If Audit Trail group policies are installed using an ADMX template instead of the plugin that the Auditing installer uses, the group policies are installed in this location in GPOE:

Computer Configuration > Policies > Administrative Templates Policy definitions (ADMX files) > Centrify Audit Trail Settings

All of the Audit Trail group policies are located in this folder, including the **Set global audit trail targets** policy.

Audit Trail Settings

There are two locations of audit trail settings:

- Some group policies are located under **Computer Configuration > Centrify Audit Trail Settings**. These policies are provided by the Centrify snap-in extension.
- Additional group policies are located under **Computer Configuration > Administrative Templates > Centrify Audit Trail Settings**. They are in an ADMX template.

Delinea

Audit Trail Snap-In Policies

For the policies located in Computer Configuration > Centrify Audit Trail Settings:

- There is a subfolder for each category of items that generate an audit trail. For example, Audit Analyzer Settings, Audit Manager Settings, and so forth.
- For global settings, the Centrify Global Settings subfolder co
- Within each subfolder, there are 2 group policies:
    - Send audit trail to Audit database
    - Send audit trail to log file

  You can configure each of these group policies to "Not configured", "Enabled", or "Disabled".

Audit Trail ADMX Template Policies

For the policies located in Computer Configuration > Administrative Templates > Centrify Audit Trail Settings:

- There is one group policy for each category of items that generate an audit trail. For example, "Set audit trail targets for category "Audit Analyzer."
- For global settings, use the "Set global audit trail targets" policy.
- You can configure each of these group policies to to "Not configured", "Enabled", or "Disabled".
- If you enable one of these policies, you also need to specify the value for Audit trail targets and Audit trail targets override. The audit trail targets and targets override is set within each of the policies (they're not separate policies).

Send Audit Trail to Audit Database

Enable this group policy to specify that audit events for this component —**Audit Analyzer**, **Audit Manager**, and so on—are sent to the active audit store database.

See the **Explain** tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Send Audit Trail to Log File

Enable this group policy to specify that audit events for this component— such as **Audit Analyzer**, **Audit Manager**, and so on—are sent to the local logging facility (syslog on UNIX systems, Windows event log on Windows systems).

See the **Explain** tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Audit Trail Overrides

This setting specifies whether to override the global audit trail targets. If this parameter is set, the system uses the targets value in the current component; otherwise, the system uses the global configured value.

There are two target settings that can be overridden:

- Whether the system sends the audit trail information to DirectAudit or not
- Whether the system sends the audit trail information to the local logging system or not. On UNIX systems, the local logging system is syslog and on Windows systems it's the Windows event log.

For this setting, you specify a single numeric value to represent where the system will send the audit trail information. (Setting one value to signify two settings is called a bit mask.) The possible settings are as follows:

0	No	No	There is no override to the audit trail target of the current component. The system uses the global audit trail target value.
1	Yes	No	The system overrides just the audit trail target for DirectAudit. This capability is supported by DirectAuditversion 3.2 and later.
2	No	Yes	The system overrides just the audit trail target for the local logging system. If you're using a DirectAuditversion prior to version 3.2, this is the default setting.
3	Yes	Yes	The system overrides both the audit trail targets for DirectAuditand the local logging system. If you're using DirectAuditversion 3.2 or later, this is the default setting.

This group policy modifies the audittrail.<product\>.<component\>.overrides settings in the agent configuration file. Each category has its own setting in that file.

Audit Trail Targets

This setting specifies how to calculate where the system sends the audit trail information for a particular component if you have also set the corresponding Audit Trail Overrides setting.

There are two kinds of audit trail targets that can be specified:

- Whether to enable the DirectAudit audit trail target for the component or not.
- Whether to enable the local logging system audit trail target or not. On UNIX systems, the local logging system is syslog and on Windows systems it's the Windows event log.

For this setting, you specify a single numeric value to represent which audit trail targets are enabled for the component. (Setting one value to signify two settings is called a bit mask.) The possible settings are as follows:

0	No	No	Neither the DirectAudit nor the local logging target are enabled for the component. This is the default setting for the group policy
1	Yes	No	Enable only the DirectAudit audit trail target for the component. This capability is supported by DirectAudit version 3.2 and later.
2	No	Yes	Enable only the local logging audit trail target for the component.
3	Yes	Yes	Enable the audit trail targets for both DirectAudit and the local logging system.

The system calculates the final audit trail targets for a component based on the following information:

- If the Audit Trail Targets Override is not specified, the system uses the global audit trail target value
- If Audit Trail Targets Override is specified, for each target (DirectAudit and local logging), whether the audit trail information will be sent to this target is determined by the following:
  - If the setting is not overridden in with Audit Trail Targets Override, the system uses the global audit trail target value
  - If the target is overridden by Audit Trail Targets Override and enabled by Audit Trail Targets, the system sends the audit trail information to this target

This group policy modifies the audittrail.<product\>.<component\>.targets settings in the agent configuration file. Each category has its own setting in that file.

Centrify Audit Settings

Centrify Auditing and Monitoring Service group policies are located in the following subfolders:

- Common Settings—Contains policies pertaining to the audit installation. See Common Settings for details about the policies in this node.
- Collector Settings—Contains policies pertaining to the collector service. See Collector Settings for details about the policies in this node.
- DirectAudit advanced monitoring—Contains policies pertaining to advanced monitoring configuration. See DirectAudit advanced monitoring for policy details.
- UNIX Agent Settings—Contains sub-nodes for policies pertaining to the Centrify Agent for *NIX. See UNIX Agent Settings for details about the policies in these sub-nodes.
- Windows Agent Settings—Contains policies pertaining to user lists used by the Centrify Agent for Windows. See Windows Agent Settings for details about the policies in this node.

Use the group policies under **Common Settings** to configure basic operations for the auditing service.

Installation

Use the Installation group policy to specify which installation agents and collectors are part of. By enabling the Installation group policy, you can prevent local administrators from configuring a computer to be part of an unauthorized installation.

> **Note**: After applying the settings through the "Centrify Auditing and Monitoring Service Settings" group policy, you must restart the target agent machine(s) for the policy to take effect.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab of the dialog box, select **Enabled**.
3. Click **Browse** to select the installation you want to secure, then click **OK**.

See the *Auditing Administrator's Guide* for more information about installing and managing installations of the auditing infrastructure.

Set the Match Order of Audit Store

When the audit and monitoring service is running on a system and it needs to connect to an audit store, you can use this policy to specify the search precedence.

If you set this policy to Enabled, then the agent looks for an audit store based on the Active Directory sites. If the search fails, then the agent looks for an audit store based on the subnet.

By default, this policy is not enabled, which means that audit stores are chosen first based on their subnet addresses and then based on the Active Directory sites.

Set Maximum Missed Status Update Tolerance

Use the Set maximum missed status update tolerance group policy to specify how many times the auditing agent will fail to connect to a collector before sending a notification that the agent is not joined to a collector. The interval between attempts is 5 minutes.

This group policy modifies the agent.max.missed.update.tolerance setting in the agent configuration file.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. Click **Edit**.
3. Select **Enabled**.
4. Enter the value. For example, enter 3 if you would like the agent to notify you after 3 failed attempts to join a collector.
5. Click **OK**.

If this group policy is Disabled or Not Configured the default value is 4.

This group policy can be used with the DirectAudit Daemon Settings group policy which allows you to specify the amount of time, in seconds, that the agent waits during each connection attempt before it determines that it cannot connect to a collector.

Set the Preferred Audit Store

Use this group policy to specify the preferred audit store that auditing will use in the event that your UNIX or Linux computer has IP addresses that match the criteria for multiple audit stores.

If you have this type of installation and you do not enable this policy and specify the preferred audit store, the collector may not connect to the correct audit store.

This group policy modifies the parameter preferred.audit.store in the agent configuration file.

Set Video Capture Auditing of User Activity

Use the Set video capture auditing of user activity group policy to specify any agents for which you want to change the video capture settings. This setting can be useful in cases where the user output should not be recorded because of security audit rules. For example, if you have enabled video capture auditing for your entire auditing installation, you can disable video capture for one or more specific agents.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab of the Properties dialog box, select **Enabled**.
3. In the Set video capture auditing section, select one of the following options:
    - **Enable Video Audit**: Select this option to turn on video capture. This setting overrides your installation-wide video capture setting.
    - **Disable Video Audit**: Select this option to turn off video capture. This setting overrides your installation-wide video capture setting.
    - **Use Installation-Wide Setting**: Select this option to make sure that this agent uses the same setting as what you have set for the entire auditing installation.
4. Click **OK** to save the change.

Use the Host Name Specified by the Agent

Enable this group policy to display the real host name of audited computers in the Audited Systems node in Audit Manager instead of the host name resolved by the collector through DNS.

This configuration parameter is useful in configurations where the DNS servers used by the collectors cannot reliably resolve host names from IP addresses. The most common scenarios that might require you to use this configuration parameter are when the agents are in a virtual environment using network address translation (NAT) or in a perimeter network outside of a firewall.

If this group policy is enabled, the host name for the agent is determined by the agent. If this group policy is not enabled, the collector determines the agent's host name based on its IP address. If this group policy is not configured, this setting will be disabled by default.

This group policy modifies the agent.send.hostname setting in the auditing configuration file.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. Click the Edit policy setting link above the policy's Description.
3. Select **Enabled**.
4. Click **OK**.

Collector Settings

Use the group policies under Collector Settings to configure the collector service.

Do Not Audit Output of Specified UNIX Commands

Use this group policy to specify one or more UNIX commands whose output you do not want to save to in the audit store database.

You can use this group policy to prevent the output from specific UNIX command that you do not want to capture or review from being saved. For example, common UNIX commands, such as the "top" and "tail" commands, might display output that you do not want to capture and store for auditing purposes. To prevent auditing the output for these types of commands, enable this group policy, click Add, then type the command.

The command string you specify must be an exact match. For example, to prevent auditing output of "cat filename", you must specify "cat filename" as the command string in this group policy.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **Add**, type the exact command you want to skip for auditing purposes, then click **OK**.
4. Repeat Step 3 for each command to skip when auditing session activity until you are finished adding commands, then click **OK**.

DirectAudit Advanced Monitoring

Use the following group policies to generate advanced monitoring for program and process execution on audited machines.

Enabling these group policies will allow you to generate reports that monitor programs and process that are run individually, as part of a script, or within other commands.

You can also configure a file monitor report which details user interaction with sensitive files.

> **Note**: You must first enable the group policy, Enable advanced monitoring to enable any of the other Advanced Monitoring policies.

Delinea

Enable Advanced Monitoring

Use this group policy to enable Advanced Monitoring.

If this policy is **Not configured**, by default, Advanced Monitoring is not enabled.

Set Monitor of Program Execution for Audit Sessions

Use this group policy to enable recording for all programs executed in an audited session. You can export these monitoring events when reviewing a session and they are also recorded in the Detailed Execution reports.

If this policy is **Not configured**, by default, this feature is not enabled.

This group policy modifies the event.execution.monitor parameter in the agent configuration file.

Delinea

Set Monitored Programs List

Use this group policy to specify a list of programs that will generate an audit trail event when executed by users.

If you enable this policy, all users executing the listed programs will generate an audit trail event, whether they are audited or not, unless the user is specified in Set skip users for monitored program executions.

Note that all commands must be specified with full paths.

If this policy is set to **Disabled** or **Not Configured**, by default, no executed programs will generate an audit trail event for any user.

This policy modifies the event.monitor.commands parameter in the agent configuration file.

Set Monitoring of System Configuration Files

Use this group policy to enable monitoring of changes made to the system configuration files in the following directory trees:

- /etc
- /var/centrify
- /var/centrifyda
- /var/centrifydc

By default, if this policy is set to **Not configured**, or if you enable this policy, all changes made to these system configuration files **will** be monitored.

Set Processes that are Skipped for System Configuration File Monitoring

Use this group policy to specify programs that modify configuration files which you do not want to be monitored when Set monitoring of system configuration files is enabled.

When you enable this policy, you can specify a list of trusted programs that can modify any system configuration files or directories without causing an audit trail event.

If this policy is **Not configured**, /usr/sbin/daspool is skipped by default, along with all adclient and dad processes and subprocesses.

Set Skip Users for Monitored Program Executions

Use this group policy to specify a list of users who can run programs and commands without generating an audit trail event.

Users listed in this policy can run commands without generating an audit trail, even if those commands are listed in Set monitored programs list.

If this policy is **Disabled** or **Not configured**, by default, all users will generate an audit trail event when executing monitored commands.

This policy modifies the event.monitor.commands.user.skiplist parameter in the agent configuration file.

Set Users that will be Skipped for Program Execution Monitoring

Use this group policy to specify a list of audited users that will not generate an audit trail event record, for use in Detailed Execution reports, when they execute programs listed in Set monitored programs list when it is enabled.

If this policy is **Not configured**, by default, no users are added to this list.

Set Users Who will be Skipped for System Configuration File Monitoring

Use this group policy to specify a list of users who can modify any system configuration file and directory without generating an audit trail event when Set monitoring of system configuration files is enabled.

If this policy is set to **Not configured**, by default, only root is added to this list.

This policy modifies the event.file.monitor.user.skiplist parameter in the agent configuration file.

UNIX Agent Settings

Use the group policy under **UNIX Agent Settings** to set auditing configuration options.

Add centrifyda.conf Properties

Use this group policy to specify any configuration parameters you want to add to the centrifyda.conf configuration file. You can specify any configuration parameter name and its value by using this group policy.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **Add**, type a property name and a property value, then click **OK**.

   For example, to change the configuration parameter autofix.nss.conf from the default value of true to false, you would type the following:

   - Property name: autofix.nss.conf
   - Property value: false
4. Repeat Collector Settings for each configuration parameter you want to set until you are finished adding property values, then click **OK**.

In typing property names and values, you should note that the agent does not perform any validation or error checking. If you specify an invalid property name or value, the parameter and value are added to the configuration file as entered. In most cases, invalid parameter names are simply be ignored. However, an invalid parameters value might cause unexpected problems when the auditing service runs.

Additional group policies for UNIX Agent Settings are organized under the following subnodes:

- **DirectAudit Daemon Settings**—Contains policies that pertain to the auditing service dad process. See DirectAudit Daemon Settings for details about the policies in this sub-node.
- **DirectAudit NSS Settings**—Contains policies that pertain to authentication requests that are processed or ignored by the Centrify name service switching (NSS) module. See DirectAudit NSS Settings for details about the policies in this sub-node.
- **DirectAudit Shell Settings**—Contains policies that pertain to the audited shell (cdash). See DirectAudit Shell Settings for details about the policies in this sub-node.
- **LRPC2 Client Settings**—Contains policies that pertain to LRPC2. See LRPC2 Client Settings for details about the policies in this sub-node.
- **Spool Disk Space Settings**—Contains policies that pertain to offline database settings. See Spool Disk Space Settings for details about the policies in this sub-node.

Delinea

Enable DirectAudit Session Auditing Properties

Use this group policy to enable and disable DirectAudit session auditing.

DirectAudit Daemon Settings

Use the group policies under DirectAudit Daemon Settings to control operations for the auditing service.

Set Allow to Dump Core

Use this group policy to specify whether the dad process is allowed to dump core. If this group policy is enabled, the dad process is allowed to dump core. If this group policy is disabled or not configured, the dad process is not allowed to dump core.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab of the Properties dialog box, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dad.dumpcore setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Audit Level of Ignored User

Use this group policy to specify the audit level of users who are on the ignored user list. Values that you can set in this policy are:

- 0 — Audit if possible.
- 1 — Do not audit.

If this group policy is disabled or not configured, a default value of 0 is used, meaning that the audit level is "audit if possible." If you enable this group policy is enabled, you can specify a value of 0 or 1.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Set the ignored user audit level to 0 or 1.
4. Click **OK** to save settings in this policy.

This group policy modifies the `user.ignore.audit.level` setting in the `/etc/centrifyda/centrifyda.conf` configuration file.

Set Cache Live Time

Use this group policy to specify the length of time entries should remain valid in the name service cache. You can specify the maximum number of seconds cached query result should be available in the cache. This policy is applicable only if the Set cache the query results policy is enabled.

If this group policy is disabled or not configured, a default value of 600 seconds is used. If this group policy is enabled, you can specify the number of seconds.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds that cached information remains valid.
4. Click **OK** to save settings in this policy.

For example, to increase the number of seconds that query results are available in the cache on an audited computer, enable this policy and specify a value of your choice that is greater than 600 seconds.

This group policy modifies the cache.time.to.live setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Cache the Query Results

Use this group policy to specify whether the dad process caches name service query results about users and groups.

- If this group policy is disabled, query results are not saved and must be retrieved whenever they are needed.
- If this group policy is enabled or not configured, the dad process stores query results—for example, from user lookup requests—in memory for better performance.
- If this group policy is enabled, you can use the Set max cache size and Set cache live time policies to control the number and duration of entries in the cache.
- If this group policy is enabled, you can also use the daflush command to clear the cache manually when you want to ensure you get updated information. For example, if you remove the UNIX Login role for an Active Directory user, some information for that user might remain in the cache and be returned when you run a command such as getent passwd. You can run daflush to ensure that the user is removed completely from the local computer cache, including the auditing name service cache.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the cache.enable setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Check NSS Configuration File Timeout

Use this group policy to specify how frequently (in seconds) the dad process checks the /etc/nsswitch.conf file for changes.

If this group policy is disabled or not configured, a default value of 60 seconds between checks is used. If this group policy is enabled, you can specify the number of seconds between checks.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds between checks.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.timer.monitor.nss.conf setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Client Idle Timeout

Use this group policy to specify how long (in seconds) the dad client can be idle before timing out. If this group policy is disabled or not configured, a default value of 1800 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds that the dad client can be idle before timing out.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.client.idle.timeout setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Codepage of Audit Client

Use this group policy to specify the code page used for character encoding by the auditing service. Supported values are UTF8 and ISO8859-1.

If this group policy is disabled, not configured, or set to a value that is not supported, a default code page of UTF8 is used. If this group policy is enabled, you can specify a supported code page.

This group policy modifies the lang_setting setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Connect to Collector Timeout

Use this group policy to specify the amount of time, in seconds, the agent waits during each connection attempt before it determines that it cannot connect to a collector.

If this group policy is disabled or not configured, the default value is 60 seconds. This group policy modifies the dad.connect.collector.timeout configuration parameter.

You can use this parameter with the Common Settings group policy which allows you to specify the number of unsuccessful attempts that the agent can make to connect to a collector before notifying the user that it is not connected to a collector.

Set Fix NSS Configuration File Automatically

Use this group policy to specify whether to enable the dad process to fix /etc/nsswitch.conf automatically if anything goes wrong.

If this group policy is disabled, /etc/nsswitch.conf is not updated. If this group policy is enabled or not configured, /etc/nsswitch.conf is updated automatically by the dad process.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the autofix.nss.conf setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Max Cache Size

Use this group policy to specify the maximum number of entries that can be stored in the name service cache. Entries store query results about users and groups. This group policy is applicable only if the Set cache the query results group policy is enabled.

If this group policy is enabled, the query results are stored in memory up to the value that you specify, resulting in better performance. If this group policy is disabled or not configured, a default value of 80,000 entries is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the maximum number of entries to cache.
4. Click **OK** to save settings in this policy.

This group policy modifies the `cache.max.size` setting in the configuration file `/etc/centrifyda/centrifyda.conf`.

Set Resource Monitor Check Interval

Use this group policy to specify how often (in seconds) the resource monitor checks dad resource usage.

If this group policy is disabled or not configured, a default value of 600 seconds is used. If this group policy is enabled and set to 0 seconds, monitoring is disabled.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds for the interval.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.timer setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Resource Monitor CPU Limit

Use this group policy to specify the maximum percentage of CPU cycles that dad can consume.

If this group policy is disabled or not configured, a default value of 50 percent is used. If this group policy is enabled and set to 0 percent, dad CPU usage is unlimited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the maximum CPU usage percentage.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.cpulimit setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Resource Monitor CPU Limit Tolerance

Use this group policy to specify (in seconds) how long the maximum percentage of dad CPU cycles can be exceeded before dad is restarted. If this group policy is disabled or not configured, a default value of 5 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds that the maximum percentage of dad CPU cycles can be exceeded.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.cpulimit.tolerance setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Resource Monitor File Descriptor Limit

Use this group policy to specify the maximum number of file descriptors that dad can open.

If this group policy is disabled or not configured, a default value of 1024 is used. If this group policy is enabled and set to 0, the number of file descriptors is unlimited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the maximum number of file descriptors.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.fdlimit setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Resource Monitor Memory Limit

Use this group policy to specify the maximum number of bytes that can be allocated to dad.

If this group policy is disabled or not configured, a default value of 104857600 bytes (100 MB) is used. If this group policy is enabled and set to 0, dad memory allocation is unlimited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the maximum number of bytes that can be allocated to dad.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.memlimit setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Resource Monitor Should Restart dad

Use this group policy to specify whether the resource monitor should restart dad if resource usage exceeds the limits set in other group policies or configuration parameters.

If this group policy is enabled, dad is restarted if resource usage exceeds specified limits. If this group policy is disabled or not configured, dad is not restarted if resource usage exceeds specified limits.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dad.resource.restart setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Seal Over a Secure GSSAPI Connection Collector

Use this group policy to specify whether the auditing service seals network communications with the collector using a secure GSSAPI connection.

If this group policy is enabled or not configured, the network connection is sealed and cannot be read. If this group policy is disabled, the connection is not sealed and is human-readable.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dad.gssapi.seal setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Sign Over a Secure GSSAPI Connection with Collector

Use this group policy to specify whether the auditing service signs network communications with the collector over a secure GSSAPI connection.

If this group policy is enabled or not configured, the network connection is signed. If this group policy is disabled, the network connection is not signed.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dad.gssapi.sign setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Soft Limit of Open Files

Use this group policy to specify the number of file descriptors that can be used for audited sessions.

For some UNIX platforms, such as Solaris, the default number of available file descriptors for each process is insufficient of auditing sessions, because the Centrify Agent requires two descriptors per session.

Use this policy to increase the number of file descriptors available.

This policy modifies the dad.process.fdlimit parameter in the agent configuration file.

Set Update Agent Status Timeout

Use this group policy to specify how often (in seconds) the agent status in the audit store database is updated.

If this group policy is disabled or not configured, a default value of 300 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds between agent status updates.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.timer.update.agent.status setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Verification of Spool Disk Space Timeout

Use this group policy to specify the number of seconds between checks of disk space when the disk space reserved for offline storage is less than the percentage specified in the Set minimum percentage of disk space group policy. At each check, a warning message is written to the log file.

If this group policy is enabled, disk space is checked at the interval that you specify. If this group policy is disabled or not configured, a default value of 360 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds between disk space checks.
4. Click **OK** to save settings in this policy.

This group policy modifies the dad.timer.diskspace setting in the configuration file /etc/centrifyda/centrifyda.conf.

Direct Audit NSS Settings

Use the group policies under **DirectAudit NSS Settings** operations for the name switching service.

Override Audit Level for a List of Users

Use this group policy to specify individual user names and audit levels or a file that contains the list of user names for which you want to override the default audit level. For more information about the how this group policy affects user auditing in classic and hierarchical zones, see the discussion of the nss.user.override.userlist parameter in the *Configuration and Tuning Reference Guide*.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type each user name and audit level using the following format:

   user_name[:*audit_level*]

   Alternatively, you can type the name of a file that contains a list of user names and audit levels.
4. Click **OK** to save your settings.

Set Audit Level for Conflict User

Use this group policy to specify the audit level to use if there is a conflict caused by a user being included in the ignores users list and having a use_sysrights audit level defined.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Select the audit level to use when there is a conflicting audit level defined for a user.
4. Click **OK** to save your settings.

Set Audit Level for Users Listed in uid.ignore

Use this group policy to specify the audit level for users who are listed in the user.ignore or uid.ignore file. For more information about the how this group policy affects user auditing in classic and hierarchical zones, see the discussion of the nss.user.override.auditlevel parameter in the *Configuration and Tuning Reference Guide*.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Select the audit level to use for users listed in the ignored user list.
4. Click **OK** to save your settings.

Set Ignored Programs

Use this group policy to list the programs that should not look up account information in Active Directory. If this group policy is not enabled or not configured, the following programs that are used for local account management are ignored by default:

useradd

userdel

adduser

usermod

mkuser

rmuser

chuser

If you enable this group policy, you must specify the list of programs to be ignored separated by spaces.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type program names separated by spaces.
4. Click **OK** to save your settings.

Set No-Login Shells

Use this group policy to specify the shells that are treated as no-login shells.

If this group policy is disabled or not configured, the shells /sbin/nologin and /bin/false are treated as no-login shells. If this group policy is enabled, specify one or more shells in a space-separated list.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type one or more shell names, separated by spaces, in the **No-login shells** field.
4. Click **OK** to save your settings.

This group policy modifies the nss.nolgin.shell setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Override Audit Level for Non-Hierarchical Zone Users

Use this group policy to specify the default audit level to use if a specific audit level is not defined for users in a classic zone. For more information about the how this group policy affects user auditing in classic zones, see the discussion of the nss.alt.zone.auditlevel parameter in the *Configuration and Tuning Reference Guide*.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Select the default audit level to use in classic zones.
4. Click **OK** to save your settings.

DirectAudit Shell Settings

Use the group policies under **DirectAudit Shell Settings** to configure shell operations for an audited shell.

Delinea

Defining Information Pattern in Custom Format to Obfuscate Sensitive Information

Use this group policy to specify information that is not displayed in auditing results. You specify the information to omit from display by defining a pattern in the group policy. Information that matches the pattern is not displayed in auditing results.

If this group policy is not configured or disabled, all information is displayed in auditing results. By default, this group policy is not configured.

If you enable this group policy, you must define a pattern as follows for information that is not displayed.

- Type the pattern that will not be displayed in auditing results. For example:

  nnnn-nnnn-nnnn-nnnn

- Each single character in a pattern corresponds to one character in actual session data.
- If you define more than one pattern, separate the patterns with spaces. For example:

  nnnn-nnnn A-nnnn

Supported characters in a pattern are as follows:

A	Any upper case letter.
d	Any character.
D	Any letter.
n	Any decimal digit character. e following: ~ ` ! @ # (space) $ % ^ & * (- _ = + [{] } | \ : ; ' < , > . ? /
-	Separator for exact matching in session data.
_	Separator for exact matching in session data.
(	Separator for exact matching in session data.
)	Separator for exact matching in session data.
,	Separator for exact matching in session data.
.	Separator for exact matching in session data.

This group policy modifies the dash.obfuscate.pattern setting in the centrifyda.conf configuration file.

Defining Information Pattern in Regex Format to Obfuscate Sensitive Information

Use this group policy to specify information that is not displayed in auditing results. You specify the information to omit from display by defining a regular expression in the group policy. Information that matches the regular expression is not displayed in auditing results.

If this group policy is not configured or disabled, all information is displayed in auditing results. By default, this group policy is not configured.

If you enable this group policy, you must define a regular expression as follows for information that is not displayed.

- Type a regular expression to define the information that will not be displayed in auditing results. For example:

  `[A-Z][0-9]{6}\\\\([0-9A-Z]\\\\)`

- If you define more than one regular expression, separate the regular expressions with spaces. For example:

  `[0-9]-[0-9] [a-z]-[0-9]`

This group policy modifies the `dash.obfuscate.regex` setting in the `centrifyda.conf` configuration file.

Delinea

Set Always Allowed Unix User Name List

Use this group policy to specify UNIX users who are allowed to use a session even if the computer cannot be audited due to environment setup issues.

If this group policy is disabled or not configured, root is the only user allowed to use an unaudited session. If you enable this group policy, you must specify a space-separated list of UNIX user names.

This group policy modifies the dash.user.alwaysallowed.list setting in the centrifyda.conf configuration file.

Set Audit All Invocations

Use this group policy to specify whether to audit all shell invocations.

If this group policy is **Enabled**, all login and non-login shells are audited.

If this group policy is **Disabled** or **Not Configured**:

- Only login shells and login sub-shells are audited.
- Invoked shells are not audited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the `dash.allinvoked` setting in the configuration file `/etc/centrifyda/centrifyda.conf`.

Set Audit Commands

Use this group policy to specify commands to audit.

If this group policy is enabled, you can create a command list and specify whether each command in the list is audited. Commands in the command list that have an action of **Enable** are audited by the auditing agent. Commands in the command list that have an action of **Disable** are not audited by the auditing agent.

If this group policy is disabled or not configured, commands to be audited must be configured manually on each UNIX computer.

When you add a command to the list, you must specify the full path to the command. You cannot add a link, shell, or wrapper script to the command list.

To use this group policy:

1. Double-click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **Show** to add or remove a command to the Audit Commands list.

   The Show Contents dialog box opens.
4. In the **Value Name** field, enter the full UNIX path name of the command.
5. In the **Value** field, enter "Enable" or "Disable" to specify whether to enable or disable auditing for the command.

   Click **OK** to save the changes and exit the Show Contents dialog box.
6. Click **Apply** in the Set audit commands dialog box to save settings in this policy, and then click **OK** to close the dialog box.

Set Audit STDIN Data

Use this group policy to specify whether the auditing agent captures standard input (stdin).

If this group policy is enabled or not configured, the auditing service records all session input and output, including standard input (stdin).

If this group policy is disabled, the auditing service records all session activity to standard output, but does not capture standard input data.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dash.auditstdin setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Continue Working Without dad

Use this group policy to specify whether the audited shell (cdash) continues to run if the dad process is not running.

If this group policy is enabled or not configured, the audited shell continues to run when the dad process is not running. If this group policy is disabled, the audited shell stops running when the dad process stops running, and the user is prompted to restart the dad process.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dash.cont.without.dad setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Except Auditing Password Strings

Use this group policy to specify strings that the auditing agent should ignore when capturing standard input data. For security, typed passwords are always ignored by default.

If this group policy is enabled, specify strings to ignore using regular expressions that do not include quotes. Leading and trailing spaces are ignored, spaces in the middle are not affected. For example:

```
dash.auditstdin.except: (prompt1\|prompt2)
```

will match strings like these:

```
This is prompt1:

Prompt2 asks for password:
```

If this group policy is disabled or not configured, this mandatory string pattern is applied:

```
(password[[:alnum:][:blank:][:punct:]]\*:[[:space:]]\*\$)\|(verify[[:alnum:] [:blank:][:punct:]]\*:[[:space:]]\*\$)
```

The default value is empty to ignore only the passwords that users enter.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type a regular expression that defines the string to ignore.
4. Click **OK** to save settings in this policy.

This group policy modifies the dash.auditstdin.except setting in the configuration file /etc/centrifyda/centrifyda.conf. For more information about specifying exceptions, see the comments in the centrifyda.conf file.

Set Force Audit List

Use this group policy to specify one or more session binary files to audit.

If this group policy is enabled, the binary files that you specify are audited. You can separate entries in the list of binary files by typing a space or a comma. You can escape spaces or commas in file names using the backslash character (\).

If the group policy is disabled or not configured, no binary files are audited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type one or more binary file names in the list.
4. Click **OK** to save settings in this policy.

This group policy modifies the dash.force.audit setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Not Audited ssh Command List

Use this group policy to specify a space-separated list of ssh commands that are not audited.

If the group policy is disabled or not configured, the commands scp, rsync, and sftp-server are not audited. If this group policy is enabled, the commands that you specify are not audited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Type one or more commands in the list, separated by spaces.
4. Click **OK** to save settings in this policy.

This group policy modifies the dash.ssh.command.skiplist setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Parent Process Skip List

Use this group policy to specify a list of parent processes that are not audited. If the name of a process's parent is in this list, the audited shell (cdash) will drop out without auditing.

If this group policy is disabled or not configured, the following processes are not audited by default:

sapstartsrv

gdm-binary

gdm-session-wor

kdm

sdt_shell

If you enable this group policy, you must specify a space-separated list of process names.

This group policy modifies the dash.parent.skiplist setting in the centrifyda.conf configuration file.

Set Reconnect to dad Timeout

Use this group policy to specify the number of seconds to wait after restarting the dad process before cdash attempts to reconnect to the auditing service.

If this group policy is enabled, the timeout that you specify is used. If this group policy is disabled or not configured, a default value of 1 second is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds to wait.
4. Click **OK** to save settings in this policy.

This group policy modifies the dash.reconnect.dad.wait.time setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Reconnect to dad Times

Use this group policy to specify how many times cdash attempts to connect to the auditing service after the dad process has started.

If this group policy is enabled, the number of attempts that you specify is used. If this group policy is disabled or not configured, a default value of 3 attempts is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of attempts.
4. Click **OK** to save settings in this policy.

This group policy modifies the dash.reconnect.dad.retry.count setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set Record Login Entry

Use this group policy to specify whether the auditing service should add utmp entries for the cdash pseudo terminals (pty). The setting of this group policy affects the results of whoami and who commands.

If this group policy is enabled, the auditing service adds utmp entries for cdash pty processes. Under this scenario, the whoami command in an audited shell works as expected, but the who command lists logged-in users twice.

If this group policy is disabled or not configured, the auditing service does not create additional utmp entries. Under this scenario, the whoami command in an audited shell cannot determine complete user information.

Workaround: on some operating systems, the who --lookup command works, but the who command lists users only once.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **OK** to save settings in this policy.

This group policy modifies the dash.loginrecord setting in the configuration file /etc/centrifyda/centrifyda.conf.

Set SHELL to Actual User Shell

Use this group policy to specify whether cdash sets the SHELL environment variable to the user's actual shell or to the audit shell.

If this group policy is enabled or not configured, the default value is true, and the SHELL environment variable is set to user's actual shell. If you disable this group policy, the SHELL environment variable is set to the DirectAudit audit shell.

This group policy modifies the dash.shell.env.var.set setting in the centrifyda.conf configuration file.

Set Skip Auditing Userlist

Use this group policy to specify the names of UNIX users and Active Directory users with a UNIX login who should not be audited. You can separate user names by typing a space or a comma. For example:

```
dash.user.skiplist: Mae kelly,dmorris,Booker
```

If this group policy is enabled, the users on the list are not audited. If this group policy is disabled or not configured, all users are audited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Create a list of users to audit.
4. Click **OK** to save settings in this policy.

This group policy modifies the `dash.user.skiplist` setting in the configuration file `/etc/centrifyda/centrifyda.conf`.

Show Actual User Running an Audited Command

Use this group policy to specify whether command-based auditing records will display the actual user account that executed the audited command, rather than just the run-as user account. Enable this policy to show both the run-as user account and the actual user account in command-based auditing records.

By default, this policy is not enabled, and only the run-as account used to run the privileged command is shown in auditing records. To enable this policy, set the parameter to `true`.

This group policy modifies the `dash.cmd.audit.show.actual.user` setting in the agent configuration file.

LPRC2 Client Settings

Use the group policies under **LRPC2 Client Settings** to control timeout and reconnect settings for the auditing service.

Set Contact with dad Timeout

Use this group policy to specify the number of seconds that cdash and dainfo wait before timing out while trying to contact the dad process.

If this group policy is enabled, the timeout that you specify is used. If this group policy is disabled or not configured, a default value of 30 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify the number of seconds to wait.
4. Click **OK** to save settings in this policy.

This group policy modifies the `lrpc2.timeout` setting in the configuration file `/etc/centrifyda/centrifyda.conf`.

Set Contact with dad Timeout for Rebinding Collector

Use this group policy to specify the number of seconds that dareload (-b) waits before timing out while trying to contact the dad process.

If this group policy is enabled, the timeout that you specify is used. If this group policy is disabled or not configured, a default value of 300 seconds is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. Select **Enabled**.
3. Specify the number of seconds to wait.
4. Click **OK** to save settings in this policy.

This group policy modifies the lrpc2.rebind.timeout setting in the configuration file /etc/centrifyda/centrifyda.conf.

Spool Disk Space Settings

Use the group policies under **Spool Disk Space Settings** to configure spool disk limits.

Set Maximum Disk Space for DB File Size

Use this group policy to specify maximum disk space (in bytes) to allocate to the offline storage database.

If this group policy is enabled, the file size that you specify is used. If this group policy is disabled or not configured, a default value of 0 bytes is used. A value of 0 bytes specifies unlimited file size.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify a file size.
4. Click **OK** to save settings in this policy.

This group policy modifies the `spool.maxdbsize` setting in the configuration file `/etc/centrifyda/centrifyda.conf`.

Set Minimum Percentage of Disk Space

Use this group policy to specify the minimum volume of disk space required on the partition containing the offline spool file before spooling stops.

You can set this value as a percentage of the disk space, or you can set it as an exact size. To set the value as an exact size, specify the unit value after the number value. The unit values are not case-sensitive.

You can specify the following unit values:

- B (byte)
- KB (kilobyte)
- MB (megabyte)
- GB (gigabyte)
- TB (terabyte)

The default value for this group policy is 10 percent of disk space.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify a value.
4. Click **OK** to save settings in this policy.

This group policy modifies the spool.diskspace.min parameter in the agent configuration file.

Delinea

Set Soft Limit Percentage of Disk Space

Use this group policy to specify the minimum volume of disk space that should be available for the offline storage file before warnings are posted to the log file. If available disk falls below the level specified in this group policy, a warning is logged and auditing will continue until disk space falls below the level specified in the Set minimum percentage of disk space group policy.

You can set this value as a percentage of the disk space, or you can set it as an exact size. To set the value as an exact size, specify the unit value after the number value. The unit values are not case-sensitive.

You can specify the following unit values:

- B (byte)
- KB (kilobyte)
- MB (megabyte)
- GB (gigabyte)
- TB (terabyte)

If this group policy is enabled, the volume that you specify is used. The default value is 12 percent.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify a value.
4. Click **OK** to save settings in this policy.

This group policy modifies the spool.diskspace.softlimit parameter in the agent configuration file.

Set Threshold Percentage of Disk Space to Reset Log State

Use this policy to specify a threshold percentage of disk space that is added to the minimum percentage of disk space (set in the Set minimum percentage of disk space group policy) that determines when the information/warning/error log state is reset. Message logging resumes only after the log state is reset.

When disk space drops below the minimum percentage (for example, 10%), a warning is logged. Additional warnings are not logged until disk space has risen above the minimum percentage + threshold percentage (for example, 10% + 2% = 12%), and then drops again to below the minimum percentage (10%).

Setting a threshold percentage is useful to prevent unnecessary log messages when disk space hovers near the minimum percentage and would otherwise trigger a log message every time the minimum percentage is crossed.

If this group policy is enabled, the percentage that you specify is used.

If this group policy is disabled or not configured, a default value of 2 percent is used.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Specify a percentage.
4. Click **OK** to save settings in this policy.

This group policy modifies the spool.diskspace.logstate.reset.threshold setting in the configuration file /etc/centrifyda/centrifyda.conf.

Windows Agent Settings

Use the group policies under **Windows Settings** to configure settings for agents on audited Windows computers.

Allow Selected Administrative Users to Stop the Auditing Service

Use this group policy to specify which users and groups can stop the auditing service on a local Windows computer using the DirectAudit Agent Control Panel.

If this policy is disabled or not configured, no users or groups can stop the auditing service through the control panel.

To use this group policy:

1. Double click the group policy in the right pane of the Group Policy Management Editor.
2. On the **Policy** tab, select **Enabled**.
3. Click **Add**.
4. In the Select Users or Groups dialog, specify the users or groups who will be able to stop the auditing service using the DirectAudit Agent Control Panel.
5. Click **OK** in the Select Users or Groups dialog.
6. Click **OK** in the group policy **Policy** tab to save your changes.

Audited User List

Use this group policy to specify which users and groups are audited. When you enable this group policy, only the users and groups you specify in the policy are audited. Be aware that this group policy takes precedence over the audit level set for a role.

If this policy is not configured, all users and groups are audited.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **Add** and identify specific users and groups to audit.
4. Click **OK** to save the list of users and groups.

See the *Auditing Administrator's Guide* for more information about the effect of choosing to enable this policy, the Windows Agent Settings policy, or a combination of both policies.

Non-Audited User List

Use this group policy to specify which users and groups are not audited. When you enable this group policy, only the users and groups you specify in the policy are not audited. If this policy is not configured, all users and groups are audited. If you enable both the Audited User List and the Windows Agent Settings policies, the users you include in the Non-audited user list take precedence over the Audited user list. Be aware that this group policy takes precedence over the audit level set for a role.

To use this group policy:

1. Double click the policy in the right pane of the Group Policy Management Editor.
2. On the Policy tab, select **Enabled**.
3. Click **Add** and identify specific users and groups to exclude from auditing.
4. Click **OK** to save the list of users and groups.

See the *Auditing Administrator's Guide* for more information about the effect of choosing to enable the Audited User List policy, the Windows Agent Settings policy, or a combination of both policies.

Set Maximum Recorded Color Quality

You can use this group policy to set the maximum color quality of recorded sessions. If this group policy is disabled or not configured, a default value of Low (8bit) is used.

To use this group policy:

1. Double click the group policy in the right pane of the Group Policy Management Editor.
2. Select **Enabled**.
3. Select one of the following options:
    - Native color
    - Low (8bit)
    - Medium (16bit)
    - Highest (32bit)
4. Click **OK** to save settings in this policy.

Set Maximum Size of the Offline Data File

You can use this group policy to specify the maximum percentage of disk space that the offline data file uses. If this group policy is disabled or not configured, the default is 10%.

To use this group policy:

1. Double click the group policy in the right pane of the Group Policy Management Editor.
2. Select **Enabled**.
3. Specify the maximum disk space percentage.
4. Click **OK** to save settings in this policy.

Set Update Agent Status Timeout

Use this group policy to specify how often (in seconds) the agent status in the audit store database is updated. If this group policy is disabled or not configured, a default value of 300 seconds is used.

To use this group policy:

1. Double click the group policy in the right pane of the Group Policy Management Editor.
2. Select **Enabled**.
3. Specify the number of seconds between agent status updates.
4. Click **OK** to save settings in this policy.

Additional Group Policies for UNIX Services

Server Suite provides additional group policies that control the configuration of specific Linux, UNIX, and Mac OS X services. This chapter describes these additional group policies.

The following topics are covered:

Common UNIX Settings

Some of the **Common UNIX Settings** group policies—such as Copy Files, Sudo Rights, and Copy Files from SYSVOL—are implemented with a dynamic link library (.dll) rather than an administrative template. Policies that are implemented with .dll plugins are always available on computers where the Server Suite Group Policy Management Extension is installed.

Other Common UNIX Settings policies—such as Set crontab Entries and Specify Commands to Run—are available only after you add the centrify_unix_settings.xml or centrify_unix_settings.admx template to the Group Policy Management Extension. You can add or remove the group policies from the centrify_unix_settings administrative template independent of the policies implemented in .dll plug-ins.

> **Note**: The Centrify Agent no longer supports the ADM administrative template in versions 2016 and later. All administrative templates must be formatted in either XML or ADMX.

Copy Files

Use this group policy to automatically copy a set of one or more files from the domain controller to each Linux, UNIX, and Mac OS X computer that joins the domain.

> **Note**: For the **Origin** domain in the **Source** file, you can only list out trusted domains in the current forest.

To enable and configure Copy files:

1. Create the files to copy in either of two locations:
   - sysvol on the domain controller.
   - A shared folder

     The sysvol location is assumed to be: \\domainController\sysvol\domainName\gpdata

     If the gpdata directory does not exist, create it first. Files to copy can be text or binary.
2. Select the Group Policy Object and click Edit to open the Group Policy Object Editor.
3. Select **Computer Configuration > Centrify Settings > Common Unix Settings**, then double-click **Copy files**.
4. In **Copy file policy setting**, select **Enabled**.
5. Click **Add**, then provide the following information:
   - Select a trusted domain or type a server name. For example, select acme.com or type a name admin1.acme.com.
   - Type the name of a file to copy or click **Browse** to browse to a directory and select a file. You can only add one file name at a time. To add multiple files, you must click **Add** for each one.
   - Type the name of a directory on the Centrify-managed computer, such as, /etc.
   - Select **Use destination file ownership and permissions** to apply permissions to the file based on the directory to which it is copied or select **Specify permissions and ownership** to manually apply permissions. When you select this button, you must enter permission data in the next three fields.
   - Enter file permissions using octal notation. Use man chmod for information.
   - Enter the UID for the file owner or click **Browse** to browse Active Directory for a user. The UID of the user you select is entered in this field.
   - Enter the GID for the user's group, or click **Browse** to browse Active Directory for a group. The GID of the group you select is entered in this field.
   - Select **Copy as binary file** to copy the file as binary. By default, files are copied as text files.
6. Click **OK** to add the specified file to the list.
7. Click **Add** to add another file to be copied.
8. When you are finished adding files, click **OK** to apply the policy with the files you have selected.
9. At any time, to remove a file, select it and click **Remove**. You may also select a file and click **Edit** to make changes to the information for the file, such as where to copy it or file permissions.

   > **Note**: If you change the policy from *enabled* to *not configured*, all files are removed from the list. However, files are not removed if you change from *enabled* to *disabled*.

Copy Files from SYSVOL

Use this group policy to automatically copy a set of one or more files from the domain controller to each Centrify-managed computer that joins the domain.

> **Note**: This group policy is still supported but has been deprecated in favor of Copy Files.

The steps to enable and configure the Copy Files from SYSVOL group policy are the same as Copy Files except that the files must be located in sysvol directory on the domain controller.

The sysvol location is assumed to be:

_domainController_\sysvol_domainName_\gpdata

You can create the gpdata directory if it does not exist, then put the files you want to copy in the directory. For more information, see Copy Files.

If you change the policy from *enabled* to *not configured*, all files are removed from the list. However, files are not removed if you change from *enabled* to *disabled*.

Sudo Rights

Use this group policy to centrally control which users can run commands as another user and the specific commands that can be run as that user. This policy configures the sudoers file with the appropriate lines when a user who has this policy applied logs on. When the user logs off, the lines applied for the user are removed and the sudoers file is restored to its previous state.

> **Note**: In order to work properly, the Sudo Rights group policy requires that the sudo package, including visudo and the sudoers file, is installed on the Delinea-managed computer.

When you select **Enabled** or **Disabled** for the Sudo Rights group policy, you can then add or remove user names and commands.

You add items to the text box just as you would to the sudoers file; that is, you type entries as you want them to appear in the sudoers file.

> **Note**: It is important to use the proper syntax for entries in the sudoers file. If the syntax isn't valid, the sudo command interprets the sudoers file as corrupt and no users are allowed to run commands using sudo rights. Therefore, in addition to the **Explain** tab, which describes the sudoers grammar in Extended Backus-Nauer Format (EBNF), this policy provides several other ways to help you enter and verify the correct syntax for your entries:

- The **Sample** tab shows sample sudoers file entries.
- A right click menu provides templates for inserting alias entries, as well as the ability to browse for users.
- Validation code verifies that there are no syntax errors in your entries before writing the entries to the sudoers file.

For example, the following procedure shows you how to create a command alias (for the rm command) and how to permit a user to simulate running as root to run the /usr/sbin/backup command:

1. In the Group Policy Editor, open the Sudo Rights policy properties and select **Enabled** or **Disabled**. Right-click and select **Insert Alias > Cmnd**. The following text is inserted in the box:

   ```
   Cmnd_Alias <alias>=<command>
   ```

2. Replace <alias> with DEL and <command> with the full path to the rm command:

   ```
   Cmnd_Alias DEL=/bin/rm
   ```

3. Click **Apply** to enter the command alias and verify that the syntax is correct.

4. On the next line, enter the following:

   ```
   jsmith ALL = /usr/bin/backup
   ```

   This entry gives jsmith all privilege on the Linux, UNIX, or Mac OS X computer to run the backup command. The user, jsmith, still needs to enter a password to run this command. You can use the context menu to change the entry and remove the password requirement.

5. After the '=' sign, insert a space, then right-click and select **Insert Value > Cmnd > NOPASSWD:** and NOPASSWD: is added to the entry.

   The entry now should now look like this:

   ```
   jsmith ALL = NOPASSWD /usr/bin/backup
   ```

6. Click **Apply** or **OK** to save the entry.

When a user to whom this policy applies logs in, the appropriate lines are added to the sudoers file. For example, when the user jsmith logs on to the computer machine1, the following is added to the sudoers file:

```
jsmith ALL = NOPASSWD /usr/bin/backup

Cmnd_Alias DEL=/bin/rm
```

If any of your entries have improper syntax, you will see an error message. Click **Details** to get information about the syntax error, then click **Cancel** and make corrections.

> **Note**: The right-click context menu also allows you to browse for user names. Right-click and select **Insert Value > Browse**, then enter search criteria. Select a name and click **OK**, and that name is added to the entry. In addition, as you add aliases, they are added to the context menu. For example, if you right-click and select **Insert Value > Cmnd**, you should see the **DEL** alias that you created in the previous procedure.

For more information about using sudo and the syntax to use in the sudoers file, see the man pages for sudo and sudoers appropriate to your operating

environment.

Set crontab Entries

Use the **Set crontab entries** group policy to manage crontab entries for individual users or for an entire computer. The management of computer-level crontab entries is performed as the root user. User-specific crontab entries run under the user's account.

Select the **Computer Configuration > Centrify Settings > Common UNIX Settings > Set crontab entries** group policy to configure computer-based policies for the root user.

Select the **User Configuration > Centrify Settings > Common UNIX Settings > Set crontab entries** group policy to configure user-based policies for individual users.

Both Set crontab entries group policies are defined in the centrify_unix_settings.xml administrative template.

If you select **Enabled** for either group policy, you can then click **Show** to add or remove entries in the /etc/crontab file.

To add crontab entries to the policy, click **Add**. You can then type the entry to be added to the file using the appropriate format for the local computer's operating environment, then click **OK**.

The standard format for entries in this file is:

```
Minute Hour DayOfMonth Month DayOfWeek User Command
```

For the Minute field, the valid values are 0 through 59. For the Hour field, the valid values are 0 through 23. For the Day of the Month field, the valid values are 1 through 31. For the Month of the Year field, the valid values are 1 through 12. For the Day of the Week field, the valid values are 0 through 6, with 0 representing Sunday. An asterisk (*) can be used in any of these fields to indicate all valid values.

For the Command field, you should type the entire command line to be executed at the specified times.

For example, to remove core files every weekday morning at 3:15 am, you could type an entry similar to this:

```
15 3 \* \* 1-5 find \$HOME -name core 2\>/dev/null \| xargs rm -f
```

Specify Commands to Run

Use the **Specify commands to run** group policy to configure one or more commands to run any time a computer is rebooted and at the computer group policy refresh interval when applied to a computer, or when a user logs on and at the user group policy refresh interval when applied to user accounts.

Select the **Computer Configuration > Centrify Settings > Common UNIX Settings > Specify commands to run** group policy to configure computer-based policies that run when a computer restarts and at the computer group policy refresh interval.

Select the **User Configuration > Centrify Settings > Common UNIX Settings > Specify commands to run** group policy to configure user-based policies that run when users log on.

Both **Specify commands to run** group policies are defined in the centrify_unix_settings.xml administrative template.

If you select **Enabled** for either group policy, you can then click **Show** to add or remove commands.

To add commands to the policy, click **Add**. You can then type the commands to be added to the file using the appropriate format for the local computer's operating environment, then click **OK**.

For computers, the commands you specify should be general computer commands.

For user accounts, the commands you specify should be user-specific. The user account that is used to run the command is recorded in the $ENV variable in the RunCommand.pl script. An entry in /var/log/centrifydc.log identifies the user. For example:

The commands are invoked for user: wtest2

Linux Settings

Use the group policies under **Linux Settings** to configure the following basic settings:

- Enforce Screen Locking
- Specify Basic Firewall Settings
- Specify Network Login Message Settings

Use the policies under **Linux Settings** › Security to configure the following computer configuration settings:

- Certificate validation method
- Enable Smart Card Support
- Lock Smart Card Screen for RHEL
- Require Smart Card Login

Use the policies under **Linux Settings** › Security to configure the following user configuration settings:

- Specify applications to import system NSSDB

Enforce Screen Locking

Use the **Enforce screen locking** group policy to control the screen lock enforcement and the timee out value for all users logging on to a computer or for individual users. Select the **Computer Configuration > Centrify Settings > Linux Settings > Enforce screen locking** group policy to configure computer-based screen locking. Select the **User Configuration > Centrify Settings > Linux Settings > Enforce screen locking** group policy to configure user-based screen locking.

Both Enforce screen locking group policies are defined in the centrify_unix_settings.xml administrative template. The mechanism used to control screen locking is specific to Linux-based computers, however, so the policies are listed under the Linux Settings category.

The most common way to handle screen locking on Linux computers is through the xscreensaver program. Although the xscreensaver program has a default configuration file, this centralized configuration file is automatically overridden if users have a local .xscreensaver file in their home directory. To enforce a centralized screen locking policy, this group policy creates a directory in the user's home directory that is owned by root and places a file that is also owned by root in this directory, so that the file cannot be removed by the user. When the xscreensaver program tests to see if there is a regular file in the user's home directory and does not find it, it uses the system configuration file.

> **Note**: If the user home directory is NFS-mounted, with the root-squash option set, this policy will not work as intended because the group policy (running as root) cannot create the un-deletable $HOME/.xscreensaver directory. As a workaround, the user may manually create the .xscreensaver directory with a umask of 0700 in the user home directory on the NFS server to prevent the user from changing .xscreensaver.

If you select **Enabled** for this group policy as a computer configuration policy, you can make the policy the default screen locking behavior for all users of the computer and set the default number of minutes to wait before locking the screen, but users are free to override the default.

To enforce this policy for individual users, you should enable the screen locking policy as a user configuration policy. However, enabling the user configuration screen locking group policy prevents users from changing their screen locking parameters.

Specify Basic Firewall Settings

Use the **Specify basic firewall settings** group policy to set up a simple exclusionary firewall on targeted computers using iptables. If you select **Enabled** for this group policy, the firewall will allow all outgoing traffic but block any inbound traffic, except ssh and ping, by default. To customize the firewall settings, select **Enabled**, then click **Show** to add or remove entries.

The Specify basic firewall settings group policy is defined in the centrify_linux_settings.xml administrative template.

To modify the default behavior of the policy, click **Add**. You can then type the appropriate entries to set up the iptables using the following format:

```
Name:Type:Protocol:Port:Action
```

where

- Name is an identifying string.
- Type is either INPUT or OUTPUT (caps are mandatory). Use INPUT to block incoming requests on the specified port and OUTPUT to block the computer from sending on that port.
- Protocol should be one of tcp, udp, icmp, or all.
- Port is the port number.
- Action is either ACCEPT or DROP.

For example, to allow connections to the computer that acts as a web server:

```
HTTP:INPUT:tcp:80:ACCEPT
```

The following example would prevent the computer from sending mail:

```
SMTP:OUTPUT:tcp:25:DROP
```

When you are finished setting up the iptables, click **OK**.

This group policy does not incorporate any Linux distribution or release-specific configurations to enable broad use of the policy.

Any existing tables are purged and new tables are built from the data pushed to the computer through the group policy.

Specify Network Login Message Settings

Enable the **Specify network login message settings** group policy to display the same welcome messages for both remote and local users. This group policy creates a symbolic link between the files /etc/issue.net and /etc/issue. If you disable the policy, the symbolic link is removed and /etc/issue.net is restored, if it existed originally.

The Specify network login message settings group policy is defined in the centrify_linux_settings.xml administrative template.

Security

Use the group policies under **Security** to configure the following computer settings:

- Certificate Validation Method
- Enable Smart Card Support
- Lock Smart Card Screen for RHEL
- Require Smart Card Login

These computer configuration policies are only applicable for Red Hat Enterprise Linux and Mac OS X. See the release notes for information about the smart card manufacturers and models supported. If you are setting group policies for Mac OS X, see the *Administrator's Guide for Mac* for additional group policies available only for this platform.

Use the group policy under **Security** to configure the following user settings:

- Specify Applications to Import System NSSDB

Delinea

Certificate Validation Method

Use this group policy to configure the certificate validation method.

For **Certificate Revocation List**, select one of the following settings:

- **Off**: No revocation checking is performed.
- **Best attempt**: The certificate passes unless the server returns an indication of a bad certificate. This setting is recommended for most environments.
- **Require if cert indicates**: If the URL to the revocation server is provided in the certificate, this setting requires a successful connection to a revocation server as well as no indication of a bad certificate. Specify this option only in a tightly controlled environment that guarantees the presence of a CRL server. If a CRL server is not available, SSL and S/MIME evaluations could hang or fail.
- **Require for all certs**: This setting requires successful validation of all certificates. Use only in a tightly controlled environment that guarantees the presence of a CRL server. If a CRL server responder is not available, SSL and S/MIME evaluations could hang or fail.

Enable Smart Card Support

Use this group policy to enable users to log in with smart cards. Enabling this policy automatically enables the Group Policy Settings Enable user group policy policy.

To remove smart card support after it has been enabled, you need to set this policy to Disabled. Changing the policy to Not configured after being Enabled does not remove the smart card requirement.

Specifying the PKCS #11 Module

Optionally, after enabling this policy, you can specify the PKCS #11 module to be used by smart card components. By default, smart card components use the Centrify Coolkey PKCS #11 module. However, Coolkey does not support all smart cards so you can specify a different module if necessary by specifying the absolute path to your PKCS #11 module in **PKCS #11 Module**. For example:

```
PKCS #11 Module /usr/\$LIB/pkcs11/opensc-pkcs11.so
```

This field supports the use of the $LIB environment variable in the path, which allows a single group policy to work for 32-bit and 64-bit systems. At run time on 32-bit systems $LIB resolves to lib, while on 64-bit systems it resolves to lib64. When you specify a PKCS #11 module, the group policy sets the following parameter in the Centrify configuration file to the specified path:

```
rhel.smartcard.pkcs11.module
```

After you enable this policy, it does not go into effect until you join the computer to the domain (if not already joined) and run the adgpupdate command.

Lock Smart Card Screen for RHEL

Use this group policy to lock the computer screen when the smart card is removed from the reader. Note that the Enable smart card support policy must be enabled in order for this policy to take effect.

To remove lock screen support after it has been enabled, you need to set this policy to Disabled. Changing the policy to Not configured does not remove this feature.

After you enable this policy, it does not go into effect until you join the computer to the domain (if not already joined) and reboot the computer.

Require Smart Card Login

Use this group policy to require all users to log in with a smart card. When this policy is enabled, no users can log in to the machine simply with a user name and password.

The Enable smart card support policy must be enabled in order for this policy to take effect. After you enable this policy, it does not go into effect until you join the computer to the domain (if not already joined) and reboot the computer.

If you don't want to require smart card login for all users, you can use the Active Directory account option to require smart card login for a specific user. For example:

- In Active Directory Users and Computers select the user's account and open the Properties.
- Click the Account tab, scroll down the list of Account options and select the **Smart card is required for interactive logon** option.

Specify Applications to Import System NSSDB

Use this group policy to specify one or more locations to import the NSS database that resides in /etc/pki/nssdb. This policy synchronizes the individual NSS application databases with the system NSS database. Enabling this policy gives these applications access to the most current certificates and CRLs. Many applications, including Firefox and Thunderbird have their own NSSDB for the user. This feature enables a mapper that parses the profiles.ini file at the location you specify and imports the certificates and CRLs to the location specified in the profile.

If you are using Firefox, you must run Firefox at least once before enabling this policy. Firefox creates the user-specific preference folder on first usage.

Enable this policy and click the **Add** button to specify the application directory in which to import the system NSS database. For each application, enter the location of its profiles.ini file. The entry must be in relation to the home directory of the user; that is, the path should start with ~/. For example, the entry for the default location of the Firefox profiles.ini file would be ~/.mozilla/firefox.

To discontinue using this policy after it has been enabled, you need to set it to Disabled. Changing the policy to Not configured does not discontinue the import.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

SSH (Secure Shell) Settings

Use the group policies **SSH Settings** to manage different aspects of secure shell (ssh) authentication. The SSH Settings group policies are defined in the centrify_unix_settings.xml administrative template.

When you set SSH Settings group policies, parameters are set in the secure shell configuration file, /etc/centrifydc/ssh/sshd_config, not in the regular configuration file. You might have other ssh configuration files stored in other default locations, depending on the operating system. The service first checks the /etc/centrifydc/ssh directory for configuration files, then looks for the configuration file in the /usr/local/etc directory on AIX computers, and /etc/ssh on AIX, SunOS, IRIX/IRIX64, and Linux computers.

Add sshd_config Properties

Use this group policy to configure secure shell properties defined in the sshd_config file by group policy.

There are two settings for this group policy:

- If the group policy is **Enabled**, you can click **Add** to add new properties as name/value pairs or edit and/or remove secure shell properties defined in the sshd_config file.
- If it is **Disabled**, or **Not Configured**, you can not add new properties as name/value pairs or edit and/or remove secure shell properties defined in the sshd_config file.

Allow Challenge-Response Authentication

Use this group policy to specify whether challenge and response authentication is allowed. Enabling this group policy sets the ChallengeResponseAuthentication option in the /etc/centrifydc/ssh/sshd_config file to yes. This setting is required to use multi-factor authentication for secure shell sessions. For more information about preparing to use multi-factor authentication, see the *Multi-factor Authentication Quick Start Guide*.

Allow Groups

Use this group policy to specify a list of groups whose members are allowed to log on through sshd. You can use wildcards (* and ?) to identify the groups to allow. Separate multiple names by spaces. Users whose primary or supplementary group membership matches any of the specified groups will be allowed to log on using a secure shell `sshd` session.

You cannot use numeric group identifiers (GID) to identify groups. By default, log in is allowed for all groups.

This group policy modifies the `AllowGroups` setting in the `/etc/centrifydc/ssh/sshd_config` file.

Allow GSSAPI Authentication

Use this group policy to allow authentication based on GSSAPI, either as the result of a successful key exchange, or through GSSAPI user authentication.

Be certain that you are using a version of OpenSSH that supports GSSAPI authentication. Otherwise, setting this policy will render the OpenSSH server unable to start.

This group policy modifies the `GSSAPIKeyExchange` setting in the `/etc/centrifydc/ssh/sshd_config` file.

Allow GSSAPI Key Exchange

Use this group policy to allow key exchanged based on GSSAPI. Note that GSSAPI key exchange does not rely on ssh keys to verify host identity.

This policy applies to protocol version 2 only.

This group policy modifies the `GSSAPIAuthentication` setting in the `/etc/centrifydc/ssh/sshd_config` file.

Allow Users

Use this group policy to specify a list of users who are allowed to log on through sshd. You may use wildcards (* and ?) to identify the users to allow. Separate multiple names by spaces.

You may also specify a host name to allow a user or users only from particular hosts. For example, mbradley@oak.com.

You may not use numerical group IDs to identify users.

This group policy modifies the Allowusers setting in the /etc/centrifydc/ssh/sshd_config file.

Deny Groups

Use this group policy to specify a list of groups whose members are not allowed to log on through sshd. You may use wildcards (* and ?) identify the groups to disallow. Separate multiple names by spaces. Log on through sshd is not allowed for users whose primary or supplementary group list matches any of the specified groups.

You may not use numerical group IDs to identify groups.

By default, log in is allowed for all groups.

This group policy modifies the DenyGroups setting in the /etc/centrifydc/ssh/sshd_config file.

Deny Users

Use this group policy to specify a list of users who are not allowed to log on through sshd. You may use wildcards (* and ?) to identify the users to disallow. Separate multiple names by spaces.

You may also specify a hostname to disallow a user or users only from particular hosts. For example, mbradley@oak.com.

You may not use numerical group IDs to identify users.

By default, log in is allowed for all users.

This group policy modifies the DenyUsers setting in the /etc/centrifydc/ssh/sshd_config file.

Enable Application Rights

Use this group policy to enable SSH application rights. Depending upon the user's role settings, this allows applications to grant rights such as password log in and allow normal shell. You configure and assign rights in zone manager.

This feature is supported in Centrify OpenSSH 4.5.4 or later. Setting this property on an unsupported version renders OpenSSH unable to start.

This group policy adds the following ServiceAuthLocation parameter to the /etc/centrifydc/ssh/sshd_config file for all computers to which the group policy object applies. It sets the path to the dzsshchk command which verifies the rights for users when they log in with SSH:

```
ServiceAuthLocation /usr/share/centrifydc/libexec/dzsshchk
```

This policy is disabled by default.

Enable PAM Authentication

Use this group policy to enable PAM authentication, account processing, and session processing. When you enable this policy, PAM authentication is implemented through the ChallengeResponseAuthentication mechanism.

Depending on your PAM configuration, enabling this policy may bypass the sshd settings of PasswordAuthentication, PermitEmptyPasswords, and PermitRootLogin without-password.

If you just want the PAM account and session checks to run without PAM authentication, then enable this policy but disable the ChallengeResponseAuthentication mechanism in sshd.

Be certain that you are using a version of OpenSSH that supports PAM authentication. Otherwise, setting this policy will render the OpenSSH server unable to start.

This group policy modifies the UsePAM setting in the /etc/centrifydc/ssh/sshd_config file.

Enable SSO MFA Properties

Use this group policy to enable multi-factor authentication for users after they authenticate through single sign-on using Centrify OpenSSH.

This group policy is only supported by OpenSSH versions 5.3.1 and later. If you attempt to enable this policy while running an earlier version of OpenSSH, the OpenSSH server will not start.

By default, this group policy is not enabled.

This group policy modifies the SSOMFA setting in the /etc/centrifydc/ssh/sshd_config file.

Match Block

You can use the Match Block group policy to add or edit match criteria so that you can match users using a variety of sub-directives.

For example, you can use this group policy if you want to set different kinds of combinations of key/value pairs to match conditions, such as the following general examples to set:

- A key/value to match a condition (key/value)
- Multiple keys/values to match a condition (key/value)
- The same keys/values to match multiple conditions (keys/values)
- Multiple keys/values to match multiple conditions (keys/values)
- Multiple conditions (keys/values) (This has the same effect as setting the policies (keys/values) individually)

For example, you could use the Match Block group policy to fulfill the following use case:

"Any user with an account login ending with *-adm will not be able to use PubkeyAuthentication"

For this example, you would set "Match User *-adm" in the match directives and set "PubkeyAuthentication no" in it's sub-directives.

The arguments to Match are one or more criteria-pattern pairs or the single token All which matches all criteria. The available criteria are User, Group, Host, LocalAddress, LocalPort, and Address.

The match patterns may consist of single entries or comma-separated lists and may use the wildcard and negation operators.

The patterns in an Address criteria may additionally contain addresses to match in CIDR address/masklen format, such as "192.0.2.0/24" or "3ffe:ffff::/32". Note that the mask length provided must be consistent with the address - it is an error to specify a mask length that is too long for the address or one with bits set in this host portion of the address. For example, “192.0.2.0/33” and “192.0.2.0/8” respectively.

Check the group policy explain text for details on which keywords can be used.

Permit Root Login

Use this group policy to specify whether and how root can log in using ssh. When you enable the policy, select one of the following options from the drop-down list:

- yes — Allow root to log in using ssh.
- without password — Disable password authentication for root. It is still possible for root to log in using another form of password authentication, such as keyboard-interactive PAM.
- forced commands only — Allow root log in with public-key authentication, but only if the command option has been enabled. All other authentication methods are disabled for root.
- no — Do not allow root to log in through ssh.

This group policy modifies the PermitRootLogin setting in the /etc/centrifydc/ssh/sshd_config file.

Set Banner Path

Use this group policy to identify a file on the Linux, UNIX, or Mac OS X computer to be sent to a remote user requesting authentication. Typically, the file contains a warning about authentication to provide legal protection to the company.

This group policy modifies the Banner setting in the /etc/centrifydc/ssh/sshd_config file.

Enable Rlogin Control SFTP

Use the Enable Rlogin Control Sftp group policy to allow remote sftp login to AIX machines when rlogin=false for the non-root users in /etc/security/user file.

This group policy overrides the rlogin=false setting in the /etc/security/user file.

Set RloginControlSftp in /etc/centrifydc/ssh/sshd_config. Default is yes. The setting term yes or no applies to whether to respect the rlogin setting or not.

- RloginControlSftp=yes, respects the rlogin=false setting, and denies the remote login. This is the default.
- RloginCongrolSftp=no, overrides the rlogin=false setting for the user, and allows remote sftp login.

Enable Rlogin Control SSH

Use the Enable Rlogin Control SSH group policy to allow remote ssh login to AIX machines when rlogin=false for the non-root users in /etc/security/user file.

This group policy overrides the rlogin=false setting in the /etc/security/user file.

Set RloginControlSsh in /etc/centrifydc/ssh/sshd_config. Default is yes. The setting term yes or no applies to whether to respect the rlogin setting or not.

- RloginControlSsh=yes, respects the rlogin=false setting, and denies the remote login. This is the default.
- RloginCongrolSsh=no, overrides the rlogin=false setting for the user, and allows remote ssh login.

Specify Authorized Key File

Use this group policy to specify the file that contains the public keys that can be used for user authentication.

If you enable this policy, specify the file in the authorized keys file box. The file specification is interpreted as an absolute path or a path relative to the user's home directory. To specify multiple files, separate each entry with a space.

The default file specification is .ssh/authorized_keys. In addition, if there are backward compatibility issues, .ssh/authorized_keys2 is checked.

Specify Ciphers Allowed for Protocol Version 2

Use this group policy to specify the ciphers allowed for SSH protocol version 2. If you enable this policy, you can add or delete ciphers to increase the speed of SSO.

Multiple ciphers must be separated by commas. If you want to add a cipher to the list, use the '+' character at the beginning of the name. If you enter the name only, you will replace the existing ciphers with the new cipher.

The order of the cipher list will determine the order that sshd uses the ciphers. For example, if you want to increase the speed of SSO, you can place the cipher, aes128-ctr, at the beginning of the list.

When this policy is disabled, the default cipher list, which is the most secure grouping, is used, but may cause delays in SSO.

To enable this group policy, you must be running Centrify OpenSSH 5.3.0 or later.

This group policy modifies the Ciphers setting in the following file: /etc/centrifydc/ssh/sshd_config.

Specify Client Alive Interval

Use this group policy to specify a timeout interval, in seconds, for requesting a response to client alive messages. If sshd does not receive a response from the client to client alive messages within the timeout interval, it sends a message through the encrypted channel requesting a response.

The default is 0, indicating that these messages are not sent to the client.

This group policy modifies the ClientAliveInterval setting in the following file: /etc/centrifydc/ssh/sshd_config.

Specify Log Level

Use this group policy to specify the log level for messages from sshd. When you enable the policy, you can select the level from a drop-down list.

The default level is INFO. DEBUG and DEBUG1 are equivalent. Logging with any of the DEBUG levels violates users privacy and is not recommended for general use.

This group policy modifies the LogLevel setting in the /etc/centrifydc/ssh/sshd_config file.

Specify Login Grace Period

Use this group policy to specify the time, in seconds, after which the server disconnects if a user has failed to log in. The default is 120 seconds.

Use 0 to specify no time limit.

This group policy modifies the LoginGraceTime setting in the /etc/centrifydc/ssh/sshd_config file.

Specify Maximum Client Alive Count

Use this group policy to specify the maximum number of client alive messages that may be sent by the secure shell daemon (sshd) without receiving a response from the client.

When the policy is enabled, the default setting is three messages.

If the threshold is reached while sshd is sending a client alive message, sshd disconnects the client, terminating the session.

This group policy modifies the ClientAliveCountMax setting in the /etc/centrifydc/ssh/sshd_config file.

Mac OS X Settings

Centrify group policies allow administrators to extend the configuration management capabilities of Windows Group Policy Objects to managed Mac OS X computers and to users who log on to Mac OS X computers. This chapter provides a high-level overview to using the group policies that can be applied to Mac OS X computers and users. For details on individual policies, see the *Administrator's Guide for Mac*.

The following topics are covered:

Group Policies and System Preferences

Windows administrators who have Mac OS X computers in their organization often want to manage settings for all of their computers and users using a standard set of tools. In a Windows environment, the standard method for managing computer and user configuration settings is through group policies applied to the appropriate site, domain, or organizational unit (OU) for computer and user accounts.

The Centrify administrative template for Mac OS X (centrify_mac_settings.xml or centrify_mac_settings.admx) provides group policies that can be applied to control the behavior of Mac OS X computers running supported versions of the Mac OS X operating system, and the configuration settings for the users who log on to those computers. By adding the administrative template for Mac OS X to a Group Policy Object, Windows administrators can access and set native Mac OS X system preferences.

This chapter provides an overview of the group policies you can enable under **Mac OS X Settings** if you add the administrative template. These group policies control the following types of Mac OS X system preferences:

- Accounts
- Appearance
- Desktop & Screen Saver
- Dock
- Saver
- Security
- Sharing
- Software Update

When you Enable a group policy in a Windows Group Policy Object, you effectively set a corresponding system preference on the local Mac OS X computer where the group policy is applied.

For example, if you enable the group policy **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Require password to unlock each secure system preference**, it is the same as opening the Security & Privacy system preference on a local Mac OS X computer, clicking **Advanced**, and setting the **Require an administrator password to access locked preferences** option.

On the local Mac OS X computer, the corresponding option is checked:

> **Note**: Not all group policies apply to all versions of the Mac OS X operating environment or all Mac computer models. If a particular system preference doesn't exist, isn't applicable, or is implemented differently on some computers, the group policy setting may be ignored or overridden by a local setting. Use the information in this chapter as a general guideline to group policies for Mac OS X. Refer to *Administrator's Guide for Mac* for detailed group-policy information for all Mac OS X versions.

Once the administrative template for setting Mac OS X group policies is installed as described below, the Windows administrator can use Group Policy Management and Group Policy Management Editor to define, link, and enforce these policies on Mac OS X computers that are joined to an Active Directory domain.

For more information about using Active Directory Users and Computers or Group Policy Management to create and link Group Policy Objects to sites, domains, or OUs, see Adding Centrify Settings to Group Policies Objects You can also refer to that section for more information about how to add administrative templates to a Group Policy Object.

Adding Mac OS X Group Policies

Centrify group policies for Mac OS X consist of two components:

- An administrative template (.xml or .admx file) that describes the policy to the Group Policy Object Editor which runs on Windows.
- A system executable and its associated configuration files that reside on the Mac and determine the policy for the local computer or for the user who is logged into the local computer and implement the policy.

Installing the Administrative Template

By default, the .xml file for Mac OS X group policies (centrify_mac_settings.xml) is installed in the C:\Program Files\Centrify\Access Manager\group policy\policy directory when you select **Group Policy Editor Extension** in the setup program. To use any of the policies, you must add centrify_mac_settings.xml to a group policy object.

Centrify provides templates in both XML and ADMX format. In most cases, it is best to use the XML template. The ADMX template file, centrify_mac_settings.admx, resides in a different directory than the .xml file.

To install the administrative template for Mac OS X group policies:

1. Create or edit an existing Group Policy Object linked to a site, domain, or OU that includes Mac OS X computers.

   For more information about creating and linking a Group Policy Object, see the Active Directory documentation or Adding Centrify settings to Group Policies Objects.

2. In the Group Policy Object Editor, expand Computer cnfiguration, then right-click Centrify Settings and select **Add/Remove Templates**.

3. Click **Add**, then navigate to the directory that contains the Centrify centrify_mac_settings.xml administrative template. By default, administrative templates are located in the local C:\Program Files\Centrify\Access Manager\group policy\policy directory.

4. Select the centrify_mac_settings.xml file, click **Open** to add this template to the list of Current Policy Templates, then click **Close**.

You should now see the administrative template for the Mac OS X group policies listed as **Mac OS X Settings** under **Centrify Settings** in the Group Policy Object Editor.

Installing the Agent and System Files

To install the Centrify Agent and the configuration files for group policy on a Mac OS X computer, run the package installer for Mac OS X and follow the instructions displayed. For more information about installing the agent or joining the domain on a Mac OS X computer, see the *Administrator's Guide for Mac*.

Enabling and Disabling Mac OS X Group Policies

Like other group policies, policies for Mac OS X users and computers are organized into categories within the Windows Group Policy Object Editor under **Computer Configuration > Centrify Settings** or **User Configuration > Centrify Settings**. These categories typically map to Mac OS X system preferences and individual policy settings map to specific system preferences settings.

Once enabled, policies get applied at the next group policy refresh interval, after the user logs out and logs back in, or after the computer has been rebooted. The description of each group policy indicates whether the policy can be applied "dynamically" at the next refresh interval or requires a re-login or a reboot.

> **Note**: The system preference updated on an individual computer must be closed, then reopened for the group policy setting to be visible.

In most cases, group policies can be Enabled to activate the policy or Disabled to deactivate a previously enabled policy. Changing a policy to Not Configured has no effect for any Mac OS X group policies. Once a group policy is set on a local computer, it remains in effect even if the computer leaves the Active Directory domain. The administrator or users with an administrative account can change settings manually at the local computer, but any manual change are overwritten when the group policy is applied.

Setting Mac OS X Computer Policies

The following table lists the categories of group policies you can set for Mac OS X computers. These group policies are in the Mac OS X administrative template (centrify_mac_settings.xml) and accessed from **Computer Configuration > Centrify Settings > Mac OS X Settings**.

802.1X Settings	Create computer profiles for wireless network authentication. The profiles you specify with these group policies are created in the Network system preferences pane.
Accounts	Control the look and operation of the login window on Mac OS X computers. These group policies correspond to Login Options in the Accounts system preference.
App Store Settings (Deprecated)	This policy was intended to control access to the App Store, however, it has been deprecated and no longer has any effect when enabled. It is provided to allow an administrator to disable the policy if it was set in an earlier version of the authentication service or the agent.
Custom Settings	Specify whether to use the Custom payload to specify preference settings for applications that use the standard **plist** format for their preference files. You can use this group policy to add keys and values to an existing preferences **plist** file.
Energy Saver	Control sleep and wake-up options on Mac OS X computers. These group policies correspond to settings in the Hardware: Energy Saver system preference.
Firewall	Control the firewall configuration on Mac OS X computers. These group policies correspond to settings in the Firewall pane of the Sharing system preference.
Internet Sharing	Manage Internet connections on Mac OS X computers. These group policies correspond to settings in the Internet pane of the Sharing system preference.
Network	Control DNS searching and proxy settings. These group policies correspond to settings in the TCP/IP and Proxies panes of the **Network** system preference.
Remote Management	Control Apple Remote Desktop access for zone users. These group policies correspond to the **Manage > Change Client Settings** options in Apple Remote Desktop.
Scripts	Deploy login scripts when an Active Directory user or local user logs on to a Mac OS X computer. You create the scripts and store them in the Active Directory domain's system volume (sysvol). They are transferred to the Mac OS X computer when the group policies are applied and executed when a user logs on.
Security & Privacy	Control security settings on Mac OS X computers. These group policies correspond to settings in the Personal: Security & Privacy system preferences.
Services	Control access to various services on Mac OS X computers. These group policies correspond to settings in the Services pane of the Sharing system preference.
Software Update Settings	Control the options for automatic software updates on Mac OS X computers. These group policies correspond to settings in the **Software Update** system preference.

For details on the individual group policies in each category and how to configure specific policies, see the *Administrator's Guide for Mac*.

Setting Mac OS X User Policies

The following table lists the categories of group policies you can set for Mac OS X users. These group policies are in the Mac OS X administrative template (centrify_mac_settings.xml) and accessed from **User Configuration > Centrify Settings > Mac OS X Settings**.

802.1X Settings	Create user profiles for wireless network authentication. The profiles you specify with these group policies are created in the Network system preferences pane.
Automount Settings	Automatically mount network share's and the Windows home directory when a user logs in.
Application Access Settings	Control the specific applications users are either permitted to use or prohibited from using. These group policies correspond to **Applications** preferences set in the Workgroup Manager.
Desktop Settings	Control the desktop and screen saver options for users on Mac OS X computers. These group policies correspond to settings in the **Desktop & Screen Saver** system preference.
Dock Settings	Control the look and operation of the Dock displayed on the user's desktop. These group policies correspond to **Dock** preferences set in the Workgroup Manager.
Finder Settings	Configure Finder commands, preferences and views.
Folder Redirection	Redirect specified folders from a network home directory to the local machine.
Import Settings	Import plist files to customize your preferences.
Login Settings	Specify frequently used items, such as applications, folders, or server connections to automatically open when a user logs in.
Media Access Settings	Control the specific media types users are either permitted to use or prohibited from using. These group policies correspond to **Media Access** preferences set in the Workgroup Manager.
Mobility Settings	Control the synchronization rules applied for users access services from mobile devices. These group policies correspond to **Mobility** preferences set in the Workgroup Manager.
Printing Settings	Specify a list of printers for a user.
Scripts (Login/Logout)	Specify login and logout scripts that run when Active Directory users log on or log out.
Security Settings	Control the secure login options for users on Mac OS X computers. These group policies correspond to settings in the **Security** system preference.
System Preference Settings	Control the specific system preferences displayed for users. These group policies correspond to **System Preferences** set in the Workgroup Manager.

For details on the individual group policies in each category and how to configure specific policies, see the *Administrator's Guide for Mac*.

GNOME Settings

The authentication and privilege elevation provide a set of GNOME group policies that control the configuration of GNOME user preferences on Linux computers. This section provides a high-level overview to using the group policies that can be applied to user preferences for the GNOME desktop environment.

This section covers the following topics:

Delinea

GNOME Desktop Preferences

GNOME is a commonly used desktop environment for Linux computers. GNOME provides a configuration system, GConf or GSettings, to store and manage GNOME user preferences. Many settings are pre-configured and stored as user preferences in the file system. The tools you use to get and set desktop preferences depend on the version of GNOME you are using. The Centrify GNOME group policies enable you to set preferences from a central location and a single interface instead of using the native tools for configuring settings. For information about setting GNOME preferences using native tools, see the documentation provided on the GNOME website.

Adding GNOME Group Policy Templates

Server Suite provides a set of GNOME group policies that implement a majority of the GNOME desktop user preferences. When enabled, these group policies use the gconftool-2 or dconf/gsettings to get and set configuration settings on Centrifymanaged Linux computers.

The Centrify GNOME group policies are defined in the centrify_gnome_settings.xml and centrify_gnome3_settings.xml template files or in centrify_gnome_settings.admx and centrify_gnome3_settings.admx template files. Group policy template files are installed automatically on the local computer if you run the setup program on a domain controller. To apply any GNOME group policy settings, you must first add one or both templates to a Group Policy Object. See Adding Centrify Policies from XML Files.

Setting GNOME Policies

Setting GNOME policies

After you add template files to a Group Policy Object, you can enable and apply the policies to computer as described in the following procedure.

To apply GNOME group policies:

1. Open the Group Policy Management Editor.
2. Open **User Configuration > Policies > Centrify Settings > GNOME Settings**.

   The right pane displays a list of folders for GNOME setting categories that correspond to the GConf settings folders on a Linux computer, and one policy, Enable GNOME group policies. By default, all group policies are set to 'Not configured'.
3. Open category folders to find the policies you want to set.

   You may need to open several layers of sub-folders. For example, to enable the policy to show hidden files in the GNOME desktop, open **desktop > gnome > file_views** to locate the **Whether to show hidden files** policy.

   You can click the **Explain** tab in any policy to review a brief explanation of the policy and its default value.
4. Double-click the policy, select **Enabled**, then click **OK** to set the policy.

   Note: In most cases, you should set all of the GNOME policies you want to deploy before performing the next step.
5. Enable the top-level Enable GNOME group policies.

   No changes to individual GNOME policies take effect until you enable this policy. This policy allows you to set GNOME user preferences exactly as you want, then implement them all at one time, rather than implement them one at a time as you set them. See Using the Enable GNOME Group Policy for more information about this policy.
6. Expand **Computer Configuration > Policies > DirectControl Settings > Group Policy Settings**.
7. Double-click **Enable user group policy**, then select **Enabled** and click **OK**.

   By default, on Linux and computers, user-based group policies are ignored until you explicitly enable them with this policy.

Verifying GNOME Policy Settings

After setting GNOME policies, you can verify the settings on any managed Linux machine by using the gconftool-2 or dconf/gsettings command.

To verify GNOME policy settings on Linux computers:

1. Set one or more GNOME group policies.
2. Enable the "Enable GNOME group policies" master policy.
3. On a managed Linux computer, run adgpupdate to apply group policies with the updates you have made.

   The agent updates group policies at a regularly specified interval. Running adgpupdate applies the new policies immediately.
4. Run gconftool-2 or use dconf/gsettings and pipe it to grep to view specific settings; for example, to see the local GNOME setting for hidden files:

   ```
   [user1@qa1 \~]\$gconftool-2 -R /desktop \lgrep -i hidden
   show_hidden_files = true
   ```

   If you are using GNOME 2, you can run gconftool-2 -R to see all of your GNOME desktop settings. For example:

   ```
   [user1@qa1 \~]\$gconftool-2 -R /desktop /desktop/gnome:
   /desktop/gnome/file_views:
   tabs_enable = true
   tabs_open_position = end
   show_hidden_files = true
   icon_theme = crux_teal
   show_backup_files = false /desktop/gnome/applications:
   /desktop/gnome/applications/component_viewer: exec = nautilus %s
   /desktop/gnome/applications/help_viewer:
   needs_term = false
   accepts_urls = true
   exec = nautilus
   ```

   To see all system settings, you can run:

   ```
   gconftool-2 -R /system
   ```

   or all desktop gnome application settings:

   ```
   gconftool-2 -R /desktop/gnome/applications
   ```

Troubleshooting GNOME Policy Settings

Troubleshooting GNOME policy settings

The GNOME group policies handle GConf settings for common applications that are installed on most Linux platforms. If one of these common applications is not installed on a user's computer, it won't be possible to set the group policies for that application. If group policy debug is enabled in the centrifydc.conf configuration file, you will see a message such as:

```
Can not get schema: user [***] gconf_key [***]
```

If none of the GNOME policies are taking effect, you should enable debug tracing and check the log file, for example, by executing the addebug command:

```
addebug set TRACE
```

In order to enable GNOME settings, sudo must be able to run without a TTY. If you see a message such as the following:

```
sudo: sorry, you must have a tty to run sudo
```

you need to edit the sudoers file on the Linux computer to allow sudo execution without a TTY.

To allow sudo execution without a TTY to allow enabling GNOME settings

1. Log in as root on the Linux computer.
2. Edit the sudoers file; for example:

   ```
   visudo
   ```
3. Find the text requiretty; for example:

   ```
   Defaults requiretty
   ```
4. Disable requiretty for all users or a specific user by using the ! symbol, as follows:

   ```
   Defaults !requiretty
   ```

   ```
   Defaults: _userName_ !requiretty
   ```
5. Save and close the file.

Using the Enable GNOME Group Policy

Because GNOME group policies affect users' desktops, it is best to apply all the policies you set at once, rather than one at a time. To support this, you can use Enable GNOME group policies as a master policy. No changes to other GNOME policies take effect until you set the master policy to Enabled. After you enable the set of policies you want to deploy, you set this policy to have all of the policies deployed at the same time.

Similarly, you can disable all previously-enabled policies at once by disabling the master policy. For example, if you want to change some existing settings, you can temporarily disable all policies, then re-enable Enable GNOME group policies when you have made all your changes.

When you disable the master Enable GNOME group policies policy, the settings on each Linux machine revert to the local GNOME settings that were in effect before you deployed group policies. The Centrify GP mapper first saves the current GNOME settings as local values on the Linux client and before it applies the Centrify GNOME settings. If you disable GNOME group policies, the Centrify GP mapper restores the local GNOME settings that were previously saved.

Creating custom GNOME Settings Through Group Policy

If you need to use group policy to configure GNOME settings that are not controlled by the default set of GNOME 3 group policies, you can use the **Custom Gnome 3 settings** group policy to do so.

If you enable the **Custom Gnome 3 settings** group policy, you specify a GNOME schema, key, and data that are implemented by the group policy. You specify the information in the group policy as follows:

- **Gnome schema:key** field: _schema id_:_keyname_

  For example:

  org.gnome.desktop.sound:theme-name

- **Data** field: _datastring_

  For example:

  freedesktop

  > **Note**: If you define custom settings in this group policy that are already defined in a default GNOME 3 group policy, the settings in the default group policy take precedence, and the settings in this group policy are not implemented.

Defining Custom Group Policies

This chapter describes how to create custom group policies and administrative templates for your Server Suite-managed systems.

The following topics are covered:

For more detailed information about creating custom group policies and administrative templates for Windows computers, see the Microsoft Web site or your Windows documentation.

Implementing Custom Group Policies

You can define your own custom group policies for Delinea-managed computers and users and add these custom group policies to existing or new Group Policy Objects. Custom group policies consist of:

- A custom administrative template (.xml) file that describes how to set the policy within the Group Policy Object Editor. For example, the Administrative Template describes the user interface presented to the administrator on Windows computer.
- A program or script that makes the appropriate settings for the computer or the user logging on. For example, you can create a Perl script that reads the group policy settings and modifies the appropriate UNIX configuration ile to reflect those settings.

Creating a Custom Administrative Template

The administrative template enables you to specify the following for a group policy:

- The policy settings, including registry settings, type of configuration (computer or user), category, and help text for the policy.
- The user interface to set the policy.
- Validation code for user-interface fields.

> **Note**: The custom Administrative Template is not strictly required if you do not need to make the settings visible and available to the Active Directory or Windows administrator, but in most cases, you should create one using a standard text editor.

Once you create your custom .xml file, you should copy the file to the C:\Program Files\Centrify\Access Manager\group policy\policy directory on a computer that has the Group Policy Object Editor (normally a domain controller) or any other accessible directory. You can then add the custom .xml file to a new or existing Group Policy Object in the same way you add any other administrative template.

Defining a Policy

Extensible Markup Language (XML) files, like a custom administrative template file, are structured documents that contain a set of supported elements enclosed in opening and closing angle (< >) brackets. The elements can be required or optional depending on the requirements of the application.

For each group policy, an administrative template provides elements to do the following:

- Place the policy in the computer configuration, in the user configuration, or in both
- Place the policy in a category
- Define the registry key entries and values to be set
- Provide explanatory text for the policy-setting page

The following example illustrates the basic file format:

```
<class type="Machine">

  <category title="DirectControl Settings"
  keynameid="CentrifyDCPolicyRegistrySettings">

     <category title="Pam Settings"
     keynameid="CentrifyDCPolicyRegistryPam">

        <policy title="Set UID conflict resolution"
        valuename="pam.uid.conflict.enabled">

        </page>

        <!--
            UI Definition
        -->
       .
       .
       .

    </page\>

     <explainpage textid="CentrifyDCPamUidConflict_Explain" />

    </policy>

    <policy title="Create k5login" valuename="pam.create.k5login">

        <valueon value="true" />

        <valueoff value="false" />

      <explainpage textid="CentrifyDCPamCreateK5Login_Explain" />

    </policy>

   </category>

  </category>

       .

       .

       .

</class>
```

Use the following keywords to define the policy:

class	Specifies the node in which to place the policy. Use one of the following with the type keyword: Machine: Computer Configuration node User: User Configuration node Both: Computer and User Configuration nodes
category	Specifies the folder for the policy. You can place a set of related policies in a single category. You can also nest categories by placing subfolders within a folder. Use title or titleid to name a category folder.

keyname keynameid	Specifies the registry setting. You can define the registry key at different levels, including category, policy, policy page or UI control, and it applies to all child levels. You can also override the setting at any child level. You should determine whether to use an existing registry key or create a new, custom key. See Defining the User Interface for a Policy for a discussion of when to use keynameid instead of keyname.
policy	Defines the policy. Use title or titleid for the display name, keyname or keynameid to specify the registry key, and page to define the property page user interface.
explainpage	Provides a page on which you can provide an explanation or instructions for the policy. The best practice is to provide a textid string for the page, and define the content (the explanatory text) of this and other strings in a separate section of the file. See Defining the User Interface for a Policy for more information.
page	Defines the property page for the policy. Use title or titleid for the page title. See Defining the User Interface for a Policy for a description of the tags you can use within page tags to define the property page.

Defining the User Interface for a Policy

You define the user interface for a group policy property page using the page tag. The template provides a number of tags that enable you to define a variety of controls, buttons, and dialogs for finding and entering Active Directory information to set group policies. Place any of the following tags within the page /page tags to define the user interface:

> **Note**: This chapter is not intended as a complete reference to the xml schema for an administrative template file, but rather shows how tags are commonly used to define a policy. For example, the current section shows how to construct the user interface to a group policy property page; specifically, it shows the tags used to create the user interface of the group policy property page. A complete reference would also show all the elements that could go into creating a dialog box, but this is not generally relevant to creating a property page and hence is not covered in this chapter.

For this type	You can specify
text	Defines a text label control. Use text or textid to define the text to be displayed in the text label.
groupbox	Groups a set of UI controls on a policy page. Use text or textid to provide a name for the box. Use keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
edittext	Creates a box in which a user can enter text. It requires the valuename keyword and value. The value should be the name used in the registry, if applicable. You can also use the following with edittext:

* text or textid to display a name for the box.
* default to display a default value when the policy is first enabled.
* keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
* maxlength *value* maximum length of the string
* charcasing to specify whether to leave the case of characters in the box as is or convert them to lowercase or uppercase. The default is to leave them as is (Normal).
* required to require a value be set.
* readonly to specify whether the value can be changed. The default is to allow the value to be changed (false).
* button to define a button to be displayed after the text control box.
* validation to define validation for user input.

numeric	Creates a numeric text box control that allows a user to adjust a numeric value up or down. It requires the valuename keyword and value. The value should be the name used in the registry, if applicable. You can also use the following with numeric:

* text or textid to display a name for the box.
* keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
* valuetype to display the type of the value in the registry setting.
* default to display a default value when the policy is first enabled.
* min *value* to set the minimum value allowed.
* max *value* to set the maximum value allowed.
* spin to define the amount to increment or decrement on each button click. The default increment is 1.
* decimalplaces to specify the number of decimal places for the value to be filled in. The default is 0.
* required to specify that the user must enter a value. The default is false, that is, the field is not required.
* validation to define validation for user input.

listbox	Provides a list view in which a user may add, remove, or edit setting values. Use dialog to associate a dialog box that enables a user to add a new entry or edit an existing entry in the list box. Specify the type of the listbox (listboxtype) to specify the kind of values the listbox generates:

* Single The box contains one column and generates a single value that is a concatenation of values from all rows separated by the separator attribute.
* Prefix The box contains one column and generates a list of registry values. The registry value name is defined by the prefix attribute and with a row number appended to the prefix name.
* Explicit The box contains two columns and generates a list of registry values. The first column contains the registry value name while the second column contains the registry value.

You can also use the following with listbox:

* text or textid to display a name for the box.
* keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
* prefix to define the prefix of the value name of the registry setting. Use this attribute with a listtype of Prefix.
* separator to separate values when the listtype is Single.
* min to set the minimum number of rows allowed.

* max to set the maximum number of rows allowed.
* sort to specify whether sorting is enabled in the list box. | | checkbox | Boolean values. This keyword requires the valuename keyword and *value*, and the valuetype. The *value* should be the name used in the registry, if applicable. You can also use the following with this checkbox:

* text or textid to display a name for the box.
* keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
* checked to set the check box to checked when the policy is first enabled. Without this keyword, the check box is not checked by default.
* valueon to define the registry setting when the check box is checked.
* valueoff to define the registry setting when the check box is not checked. | | radiogroup | Defines a set of two or more radio buttons (radiobutton) from which a user must make a single choice. This keyword requires the valuename keyword and value, and the valuetype. The value should be the name used in the registry, if applicable.

You can also use the following with radiogroup:

* text or textid to display a name for the box.
* keyname or keynameid to specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
* radiobutton to define radio buttons for the control. Use checked=true to specify the default radio button. | | radiobutton | A list of suggestions to allow the user to select or type a value. It requires the valuename keyword and value. The value should be the name used in the registry, if applicable. You can also use the following with combobox:

* textortextidto display a name for the box.
*checkedto define the default state for the radio button. The default isfalse(not checked).
*valueonto specify a value to be written to the registry when the radio button is checked. | | dropdownlist | A list of suggestions to allow the user to select a value. It requires thevaluenamekeyword andvalue. The value should be the name used in the registry, if applicable. You can also use the following attributes with dropdownlist:

* valuetypeto define the type of value in the registry setting.
*textortextidto display a name for the box.
*keynameorkeynameidto specify the registry setting. Setting the registry key at this level overrides any setting at a higher level, for example, at the category level.
*editableto specify whether the value in the dropdown list may be edited. The default isfalse(cannot be edited).
*requiredto require a value be set.
*sortto specify whether sorting is enabled in the dropdown list box.

You can use the following tags withindropdownlist:

* listitemto define an item in the drop-down list.
*validationto define validation for user input. | | button | Creates a button for a text field defined byedittext.

Use the dialog or adbrowse tags with button to define a dialog box to be shown when a user clicks the button.

You can also use the following attributes with button:

* textortextidto display a name for the box.
*valueidto identify the value returned from the dialog box that is launched by clicking the button. | | dialog | Provides a dialog box. You associate a dialog box to a button or to alistbox. Use titleortitleidto specify the title for the dialog.

You can use the following child tags to define a dialog box:

*groupboxto define a group box control in the dialog.
*textto define a text control in the dialog.
*edittextto define a text edit box control in the dialog.
*numericto define a numeric up down control in the dialog.
*listboxto define a list box control in the dialog.
*checkboxto define a check box control in the dialog.
*radiogroupto define a group of radio button controls in the dialog.
*dropdownlistto define a drop down list control in the dialog.
*validationto define the validation on the user inputs in the dialog. | | adbrowse | Provides a dialog box for browsing. You associate an adbrowse dialog box to a button or to alistbox. Use textortextidto specify the title for the dialog.

To browse Active Directory, useadbrowsetype to identify the type of browsing:

*FindADUser
*FindADGroup
*FindUnixUser
*FindUnixGroup
*FindComputer

Usemultiselectto define whether a user can select multiple search results in the Active Directory browse dialog.

Useseparatorto specify the separator for multiple results.

You can use the following child tags to define anadbrowsedialog box:

*groupboxto define a group box control in the dialog.
*textto define a text control in the dialog.
*edittextto define a text edit box control in the dialog.
*numericto define a numeric up down control in the dialog.
*listboxto define a list box control in the dialog.
*checkboxto define a check box control in the dialog.
*radiogroupto define a group of radio button controls in the dialog.
*dropdownlist` to define a drop down list control in the dialog. |

Using String IDs

When entering strings, such as text, keynames, and titles, you have the choice of using strings or string IDs. String IDs offer several advantages, such as a cleaner, more modular design, and the ability to customize the text if you plan to port to different languages.

The best practice is to put the string IDs in a 'Strings' section of the template file, which makes them easy to locate and modify in case of porting to other languages.

For example, the following segment from a template file shows how the explainpage tag specifies a string ID to attach explanatory text for a policy to the policy dialog box, while the actual text is defined in a 'Strings' section at a different place in the template:

```
- <!--

        Set login password prompt

  -->

- <policy title="Set login password prompt"
            valuename="pam.password.enter.enabled">

- <page\>

  - <edittext text="Set login password prompt"
              valuename="pam.password.enter.mesg"
      maxlength="1024" default="Password:">

    </edittext>

  </page>

__<explainpage textid="CentrifyDCPasswordPrompt_Explain" />__

  </policy>

- <!--

.

.

.

- <!--

============================================================

Strings

============================================================

__<string id="CentrifyDCPasswordPrompt_Explain">__The prompt that is displayed when an Active Directory user attempts to log in. Environment variables may be used in the form
\$VARNAME if a '\$' character is desired, escape it: \$</string>`

<string id="CentrifyDCPasswordChangeNotify_Explain">The message that is
displayed to an Active Directory user when they attempt to change their
password. Environment variables may be used in the form $VARNAME if a '$'
character is desired, escape it: \$</string>

.

.

.
```

Validation Settings

You can write validation scripts to check individual settings. The validation scripts are run after a user enters settings but before the settings are saved.

You can use any of the following languages to write validation scripts:

- VBScript
- JScript
- C#
- VB.net

Use the validation tag to apply a validation script to a setting. Use method to define the validation method name. Use param to define a parameter value to pass to the method or paramval to pass a registry setting value to the method. The validation result is returned by the method's return value. Use either dotnetscript to define a .net script (C# or VB.net), or script to define a script (VBScript or JScript) to do the validation.

The following segment from an administrative template file illustrates how to call a validation method:

```
- <validation>
    <method name="Validation.CheckUser" />
  - <dotnetscript language="C#">
    - <code>
      - <![CDATA[
public class Validation
                    {
                      public static string[] CheckUser(string value)
                      {
                        return Utility.CheckUnixNames(value, new
                       char[] { }, "Unix user name");
                      }
                    }
    ]]>
  </code>
 </dotnetscript>
</validation>
```

You place the code to call the method within a CDATA tag. Likewise, place the validation code itself within a CDATA tag, as in the following example:

```
- <dotnetscript language="C#">
   - <code>
    - <!--
       Validation Utility
      -->
    - <![CDATA[
        using System;
        using System.Text;
        public class Utility
        {
          .
          .
          .
         /// <summary>
```

```
        /// Check for a list of Unix names separated by seps

        /// </summary>

        /// <param name="value">\</param>

        /// <param name="seps">\</param>

        /// <param name="displayText"></param>

        /// <returns></returns>

        public static string[] CheckUnixNames(string value, char[]
        seps, string displayText)

        {

          .

          .

          .

        }

      }

]]>

  </code>

</dotnetscript>
```

Adding a Mapper Program to the Agent

To implement group policies for UNIX computers and users, you need to create the custom scripts or programs that modify the appropriate UNIX configuration files or settings. You can create the programs or scripts using the programming or scripting language of your choice. Most of the Centrify policies use Perl scripts and you can use those scripts for models if you choose to use Perl.

Once you create a program or script to implement a group policy, you need to:

- Place the program or script in the /usr/share/centrifydc/mappers/machine directory if it is a computer configuration group policy, or in the /usr/share/centrifydc/mappers/user/*user_name* directory if it is a user configuration group policy.
- Make the program or script an executable file.
- Use the runmappers command to test that the program or script works as expected and updates the appropriate configuration file.

By default, when you use the runmappers command, it executes all of the programs in both the /usr/share/centrifydc/mappers/machine and the /usr/share/centrifydc/mappers/user/*user_name* directories. Optionally, you can run the command to only execute your custom program. For example, if you have created an executable script called setport.pl as a UNIX computer configuration policy and placed the file in the /usr/share/centrifydc/mappers/machine directory, you could use a command similar to the following to execute the script along with the other computer configuration mapper programs and test its behavior:

```
runmappers machine map
```

> **Note**: To run the mapping programs for a user, you must specify the user's UNIX login name to identify which user's group policies should be mapped or unmapped. For example, to run the mapping programs for the UNIX user account jgarcia in the /usr/share/centrifydc/mappers/user/jgarcia directory, you could use a command similar to the following:

```
runmappers user jgarcia map
```

This section provides a brief overview of Network Information Services (NIS), including the basic advantages and limitations of using NIS to publish information. It also describes the Centrify solution for using NIS to respond to client authentication and lookup requests.

You should use this section to help you determine whether the Centrify Network Information Service (adnisd) is an appropriate solution for your organization's needs.

Introduction to the Basics of NIS

In some environments, a Network Information Server (NIS) provides centralized storage and distribution of information that needs to be known throughout the network. In a typical NIS environment, one or more NIS servers are used to centrally manage a set of database **maps** that correspond to the system configuration files that are commonly found on UNIX systems. For example, there are NIS maps that correspond to the /etc/passwd, /etc/group, /etc/hosts, and /etc/services files. The maps provide the centralized information to a given set of computers that make up a **NIS domain**.

Each **NIS map** corresponds to a specific configuration file, such as the /etc/passwd or /etc/hosts file, and consists of a set of keys and values, and a version number for the data. When computers on the network require information stored in NIS maps, they send a **NIS client request** to the NIS listening port to query the **NIS server** for the information.

When a computer needs the information stored in a NIS map, it runs the ypbind process to identify and connect to the NIS server best suited to respond to its requests. When the NIS server receives a request, it replies with the appropriate information from its set of NIS maps.

Limitations of using NIS

Although NIS can be very efficient in responding to queries for network information, it is not a secure mechanism for providing authentication and authorization services. For example:

- If NIS clients use the broadcast service to locate NIS servers on the network, intruders can easily introduce their own NIS server with their own privileged accounts. Once a client binds to the rogue NIS server, the intruder can gain access to that client and perform unauthorized operations.
- The NIS server's only security policy is the securenets setting. The securenets setting identifies which NIS clients to accept queries from. If an intruder impersonates a client that the securenets setting allows the NIS server to accept, he can download all of the NIS data. Even if an intruder fails the securenets test, he could potentially inspect all of the NIS requests and decode the data to gain access.
- If NIS is used for authentication, password hashes are sent around the network in clear text and can be easily captured and cracked, making client systems vulnerable.

Because of these security risks, in most cases, you should plan to replace any legacy NIS environment with Active Directory as the central repository of identity information and the Centrify Agent for *NIX (adclient) as the "client" requesting information. In some cases, however, if may not be practical or desirable to completely replace an existing NIS infrastructure. To handle those cases, Centrify provides its own Network Information Service (adnisd) that enables existing NIS clients to remain in place and co-exist with Active Directory.

Deciding to Maintain NIS in your Environment

Active Directory and the Centrify Agent for *NIX (adclient) provide more secure authentication, authorization, and directory services than provided by traditional NIS client-server communication. Therefore, when you install the Centrify Agent and join a domain, the Name Service Switch configuration file, nsswitch.conf, is normally modified so that account lookup requests are passed to Active Directory through the adclient process. This change to the nsswitch.conf file effectively bypasses the NIS client and server environment.

There are some situations, however, in which maintaining an ongoing or temporary NIS environment may be desirable or necessary. For example:

- If you have a legacy Network Information Server (NIS), you may have configured network information, such as netgroup or automount maps, that you want to make available in response to client requests.
- You may have applications that require access to a NIS server because they send requests directly to the NIS port and expect a NIS process to be listening there.
- You may have computers or devices, such as Network Attached Storage devices or computers with older or unsupported operating systems where you cannot install the Centrify Agent, that need access to information normally stored in NIS maps. Those computers or devices cannot join an Active Directory domain, but are capable of submitting NIS client requests. For those computers or devices, a NIS server may be the only option for providing authentication and look-up services.

If any of these scenarios apply to your organization, you may want to plan a deployment that includes the Centrify Network Information Service to complement the agent.

Using the Network Information Service

To support computers and applications that are capable of submitting NIS client requests to a NIS server, the Server Suite provides its own Network Information Service. The Centrify Network Information Service, adnisd, is an optional process that can be installed on any computer where adclient is installed.

Once installed and running, the Centrify Network Information Service functions like a standard NIS server, but it responds to NIS client requests using the information stored in Active Directory, including any information imported from passwd and group NIS maps or from /etc/passwd and /etc/group files. The Centrify Network Information Service has some of the same security limitations as a standard NIS server, but it does allow you to provide encrypted authentication and directory service to computers where adclient cannot be installed.

The Centrify Network Information Service can be useful in environments where you plan a phased migration from existing NIS servers and clients or when the environment includes legacy systems that you cannot migrate or upgrade to support the Centrify Agent for *NIX.

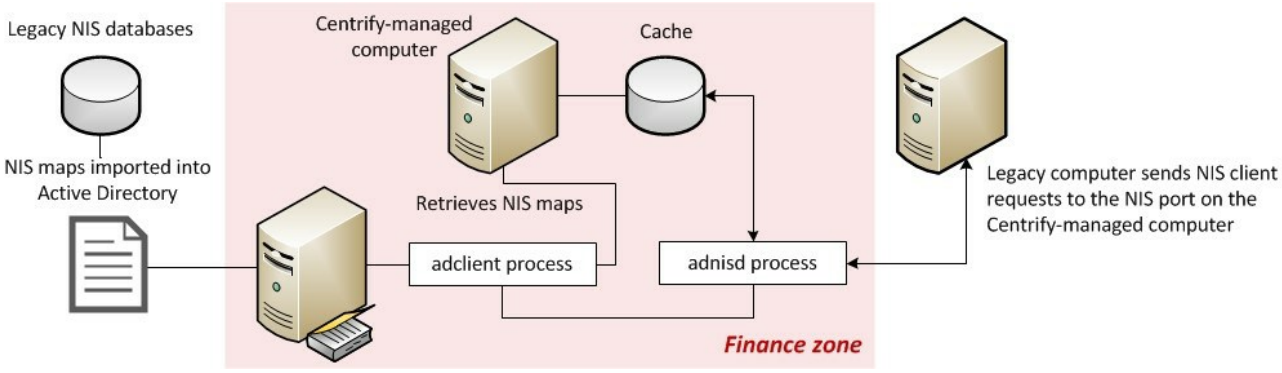
How NIS Client Requests are Processed

If you have decided to maintain a NIS environment, on either an ongoing or temporary basis, you can use the Centrify Network Information Service to replace existing NIS servers and the Access Manager console to migrate NIS map data to Active Directory.

The Centrify Network Information Service (adnisd) can run on any computer that has the adclient agent service installed. Computers that need access to the information stored in Active Directory can then be configured as NIS clients that send their NIS queries to the computer where both the adclient and adnisd service run.

When adnisd receives a request from the NIS client, it checks its local cache of map data, then responds to the client that made the request. The local cache of map data is generated from the map data adnisd receives from Active Directory.

The following figure provides a simplified view of operation.



Derived and Explicitly-defined Maps

Within the local cache, there are two types of maps: **explicitly-defined maps** and **derived maps**. Explicitly-defined maps are NIS maps imported into Active Directory from an existing NIS domain, imported from text files, or created manually using the Centrify Access Manager console. Derived maps are maps that are automatically generated from the information stored in Active Directory. Derived maps access the same data as the explicitly-defined maps using different keys. For example, the user and group maps in the local cache are not retrieved directly from Active Directory, but are generated based on the users and groups that have been enabled for the local computer's zone.

The maps derived from the zone information are passwd.byname, passwd.byuid, group.byname, and group.bygid. These automatically generated maps are placed in the local cache, and can then be used to look up or authenticate users by user name or by UID value, and groups by group name or by GID value. The Centrify Network Information Service also generates derived maps for explicitly-defined network maps that are stored in Active Directory. If adnisd finds a NIS map defined in Active Directory with a name it recognizes, such as netgroup or services, it automatically derives related maps. For example, a netgroup map will automatically generate the netgroup.byhost and netgroup.byuser maps. A services base map will generate the services.byname and services.byservicename maps.

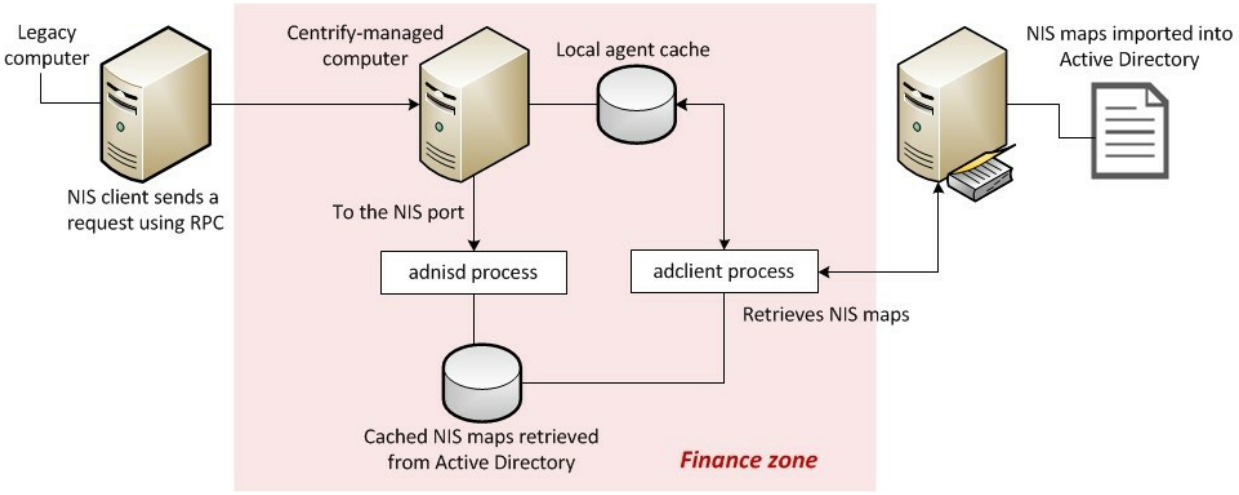
Accessing NIS Maps in the Local Cache

Periodically, theadnisdprocess connects to Active Directory through the adclient process to locate updates to explicitly-defined NIS maps. It then synchronizes the local cache of NIS map data to mirror any changes detected in Active Directory. After polling Active Directory for updates to explicitly-

defined maps, the adnisd process retrieves all users and groups in the current zone from adclient, and generates the derived maps for user and group information.

In essence, the computer where both adclient and adnisd run acts as the NIS server for the local computer's zone. The NIS clients on the network communicate with adnisd using Remote Procedure Calls (RPC) sent to the NIS port on the Centrify-managed computer. The adclient process is responsible for all communication with Active Directory and maintains its own separate cache of data from which adnisd can derive the user and group information for the zone. The adnisd process then stores all of the explicitly-defined and derived maps in its own local cache of map data (in most cases, /var/centrifydc/nis/*). Because adnisd always responds to NIS client requests using the data in its local cache, it can respond even when Active Directory is not available.

The following figure provides a simplified summary of operation.



> **Note**: The adnisd process cannot be used with any legacy NIS servers in a NIS domain. It can only be used in conjunction with Active Directory and the Centrify Agent for *NIX.

Migrating Information from Existing Maps

If you have a legacy NIS environment, you can import user, group, and network information from existing NIS servers and domains. To import the information directly from an existing NIS server, you need to be able to access the NIS server and NIS domain from the Windows computer where the Access Manager console is installed. For example, if you have configured an existing NIS server to be accessible over the Windows network using Samba or a similar program, you can connect directly to that server and NIS domain to import maps. If the NIS server and NIS domain are not accessible from the Windows computer where the Access Manager console is installed, you should export the NIS maps to text files, then import the text files.

> **Note**: Importing existing maps simply provides a mechanism for migrating existing information to the Active Directory. Once the information is imported into Active Directory, the original maps are no longer used and the Centrify Network Information Service uses Active Directory to generate the maps it needs to service authentication requests.

For more information about importing existing user, group, or network information, see Importing and managing NIS maps.

Managing Automounts without Using NIS

If your primary reason for wanting to use NIS is to manage automount information, you have the option of storing the information in Active Directory then retrieving it through the adnisd process or directly through an LDAP request that bypasses the adnisd process.

> **Note**: The automount information must be stored in Active Directory. You can then choose whether to retrieve it using the Centrify Network Information Service (adnisd) or an LDAP query.

As an alternative to using the adnisd process, you can use the optional adauto.pl script located in the /usr/share/centrifydc/etc directory to get automount data. The adauto.pl script gets mount point information directly from Active Directory using LDAP. With the adauto.pl script, you can automount home directories using the information from NIS maps without running the adnisd server process.

The adauto.pl script uses the information you store in the auto.home NIS map for the joined zone and any parent zones up the zone hierarchy from which the local computer inherits NIS map entries. Once you add the script to your automount configuration, the automounter program invokes the script and passes it the user

name of the user logging on. The adauto.pl script then uses the ldapsearch command to retrieve the mount point information from Active Directory and returns the path to the remote home directory for the user logging on. The automounter will then attempt to connect to that home directory.

To use the adauto.pl script:

1. Add the appropriate auto.home mount points to Active Directory by importing or creating automount NIS maps.

   For more information about importing existing auto.home or auto_home NIS maps, see *Importing Network NIS Maps*. For information about creating NIS network maps directly in Active Directory, see Creating new NIS Maps in Active Directory.

   For example:

   - Open Access Manager to navigate to a specific zone.
   - Expand the zone to display NIS Maps.
   - Select NIS Maps, right-click, then click New > Automount.
   - Type auto.home or auto_home as the map name, then click **OK**.
   - Select the new map, right click, then click New to add a new individual map record. For example, create a map record similar to this for all users in a zone:

     Name: * Network Path: lmrh2:/home/&
     Comments: This is the automount path for users in this zone

2. If you are managing mount points on Linux or Solaris, edit the /etc/nsswitch.conf file to change the automount entry from nis to files. For example:

```
vi /etc/nsswitch.conf
...
automount: files
```

For other platforms, such as AIX, you can skip this step.

1. Verify the adauto.pl file is available in the /usr/share/centrifydc/etc/ directory and is executable. For example:

   ls -l /usr/share/centrifydc/etc/adauto.pl
   total 1208
   -rwxr-xr-x 1 root root 1921 Sep 27 10:37 adauto.pl

2. Create a symbolic link for /etc/auto.home or /etc/auto_home to the adauto.pl file. For example, on Linux computers:

   ln -s /usr/share/centrifydc/etc/adauto.pl /etc/auto.home

   On AIX computers, create the link to /etc/auto_home:

   ln -s /usr/share/centrifydc/etc/adauto.pl /etc/auto_home

3. Edit the /etc/auto.master or /etc/auto_master file to call the /etc/auto.home file.

   For example, on Linux computers add the following line to the auto.master file:

   /export/home program:/etc/auto.home

   The specific syntax for the entry is different on different platforms. For example, not all platforms allow you to specify the program keyword in the /etc/auto.master file. For more information about the format of the entry, see the main page for auto.master. For example, on SuSE Linux, the entry should look like this:

   /export/home /etc/auto.home

   On SuSE Linux 10, the corresponding entry is:

   /export/home program /etc/auto.home

   On AIX and Solaris computers, add an entry like this to the /etc/auto_master file:

   /export/home /etc/auto_home

   On some platforms, you can invoke automount from the command line without editing the /etc/auto.master file. For example, you can invoke automount without

editing the /etc/auto.master file by running a command similar to the following on Linux:

```
automount /export/home/ program /etc/auto.home
```

Command line mount points are not supported by automount on AIX.

4. Restart the autofs process. For example, on Linux:

   ```
   service autofs restart
   ```

   On AIX:

   ```
   automount
   ```

   On Solaris 10, the automount service is managed by the service management facility, smf, under the service identifier:

   ```
   svc:/system/filesystem/autofs:default
   ```

   You can use svcadm to perform administrative actions, such as stopping and restarting the service.

Mounting Home Directories with the nosuid Option

To increase security when automatically mounting file systems, you might want to configure the auto_home or auto.home NIS map to prevent users from switching their user or group identity. You can prevent users from mounting file systems with a different user context by specifying the nosuid option.

To set the nosuid option in the auto_home or auto.home NIS map:

1. Open Access Manager to import or create a NIS map to be stored in Active Directory.
2. Expand the appropriate zone and the UNIX Data node to display NIS Maps.
3. Select NIS Maps, right-click, then click New > Automount.
4. Type auto.home or auto_home as the map name, then click **OK**.
5. Right-click the new map.
6. Click **New > Map entry** to add a new individual map record.
7. Set the fields in the map record similar to the following, to enable mounting of home directories with the nosuid option for all users in a zone:

   Name: * Network Path: homeservername:/home/&
   Options: -nosuid

   You can use a similar approach to specify other or additional mount options—such as noexec and nodev—to the map entry.

Using Executable Maps

On some platforms, local maps that have the execute bit set in their file permissions can be executed by the automount program and provided with a key to be looked up as an argument. The executable map is expected to return the content of an automount map entry on its stdout or no output if the entry cannot be determined. Direct maps cannot be made executable.

For more information about executable maps, see the main page forautomount.

Testing the Status of the Automount Service

After restarting the automount service, you can check the status of the service. For example, on Linux run the following command:

```
service autofs status
```

On all platforms, you can run the following command and check the output to verify automount operation:

```
/usr/sbin/automount -V
```

You should see output similar to the following:

```
automount: /export/home mounted automount: no unmounts
```

Running the adauto.pl script

You can run the adauto.pl script with no command-line options to manually refresh the automount NIS maps on demand. Alternatively, you can manually add the adauto.reloadtime configuration parameter to the /etc/centrifydc/centrifydc.conf file to control how frequently automount NIS maps are retrieved for the zone. If you manually add this parameter to the configuration files, you can set the value to specify that maps with a time stamp older than the specified number of minutes should be reloaded.

By default, the adauto.pl script gets automount NIS maps from the zone to which the local computer is joined. If the maps are not found in the joined zone, the script will attempt to get the maps from its parent zone of the joined zone. Alternatively, you can create the file /var/centrifydc/kset.automap and type the common name (CN) of the specific Centrify zone from which you want to load the automount NIS maps.

Testing the adauto.pl script results

After you have configured the auto.home and auto.master maps, you can test that the adauto.pl script is working by entering one of the following commands:

/etc/auto.home userid /etc/auto_home userid

This command should return the path from the auto.home or auto_home NIS map stored in Active Directory. For example:

/server/home/userid

Restarting the Automount Service

If you make any changes to the NIS maps in Active Directory, you should restart the automount service.

Distributing Automount Maps

You can create auto.master and auto.home files as NIS maps in Centrify zones and distribute them using symbolic links to the adauto.pl script. In this scenario, you can take advantage of the capability to support executable maps. Depending on your operating system, however, you might be able to take advantage of the Centrify NSS module to automatically mount home directories instead. If your operating system allows you to use the Centrify NSS module, you can add centrifydc to the automount line in the /etc/nsswitch.conf file.

In most cases, you can use the Centrify NSS module to distribute auto.home maps. You cannot use this approach, however, to distribute the auto.master map on most operating systems. For the auto.master map, your options are typically limited to doing one of the following:

- using NIS.
- using LDAP.
- using a local file.

For information about using LDAP, see "Using the Centrify LDAP proxy service" in the *Administrator's Guide for Linux and UNIX*. If you use a local file, you can use an adedit script to synchronize the auto.master map to a local /etc/auto.master file. The following example illustrates the steps to synchronize the auto.master map to a local /etc/auto.master file.

1. Add the File Copy group policy to a Group Policy Object that applies to Centrify-managed computers.
2. Enable the group policy to copy a script similar to the following to the directory /usr/share/centrifydc/mappers/machine:

```
#!/bin/sh
# Restart adedit using tclsh \
exec adedit "$0" "$@"
# Bind to an Active Directory domain \
bind -machine domain
# Select a zone context \
select_zone zone
catch  {
    select_nis_map auto.master
    set output [open /etc/auto.master w 0644]
    foreach line [gnm] {
    puts $output [regsub ":1" $line ""]
      }
    close $output
    }
}
```

By adding a script similar to this sample script to a GPO, every 90 to 120 minutes the group policy update will execute the script to read the contents of the auto.master map in Active Directory and create a local copy of the /etc/auto.master file.

You can also use this same approach to synchronize all of the maps stored in Active Directory to the local /etc directory. For example:

```
#!/bin/sh
 # Restarts using tclsh \
 exec adedit "$0" "$@"
 bind -machine [adinfo domain]
 slz [adinfo zone]
 foreach map [get_nis_maps] {
    if ([regexp "auto*" $map]) {
       slnm $map
       set output [open /etc/$map w 0644]
          foreach line [gnm] {
          puts $output [regsub ":1" $line ""]
       }
    close $output
    }
}
```

Discontinuing Use of Legacy NIS Servers

If you have existing NIS servers running on your network, you can configure your NIS clients to use the Centrify Network Information Service over time, as needed. Once you have the Network Information Service running, you can also incrementally update the NIS data that's stored in Active Directory using the Access Manager console. Any updates you make are then propagated to all of the `adnisd` servers automatically.

When you are satisfied that you have all of the appropriate NIS information stored in Active Directory and have deployed `adnisd` across the enterprise, as needed, you can then stop any remaining legacy NIS servers and complete the migration to Active Directory for secure, centralized directory service.

> **Note**: Although you can leave the standard NIS servers in place indefinitely, you should plan to migrate all of your data and NIS clients to use the Centrify Network Information Service if you want you to centralize all authentication and directory service in Active Directory. Once you have imported all of the data you need into Active Directory and configured your existing NIS clients to use the Centrify Network Information Service in the appropriate zone, you can decommission any legacy NIS servers and stop any related services.

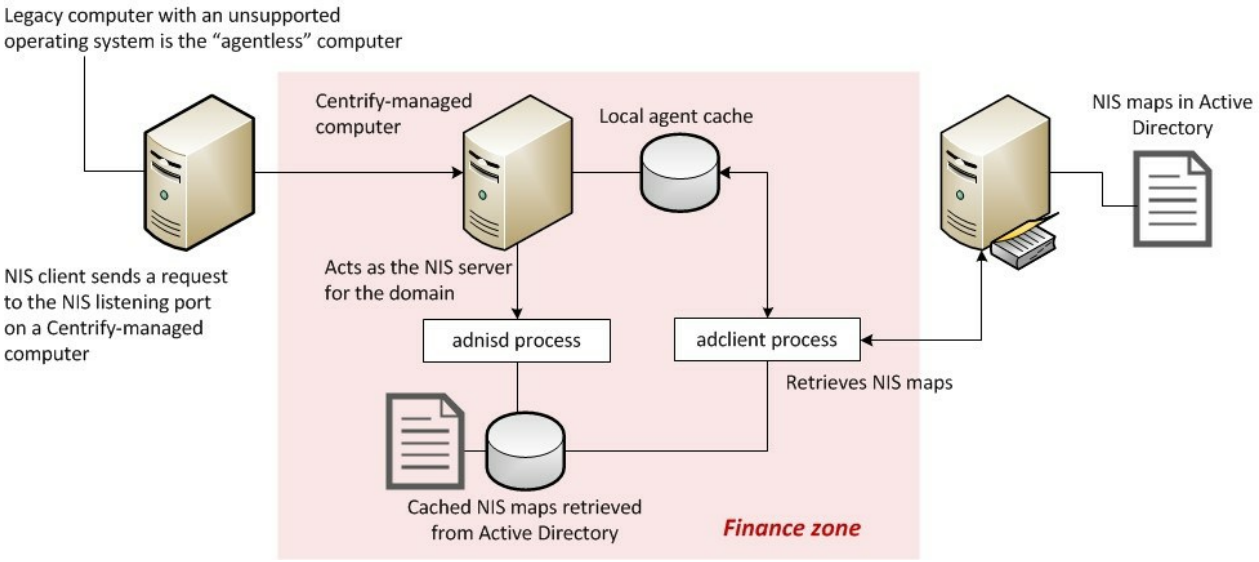
This section describes the activities that are specific to preparing your environment to handle agentless authentication and authorization. If you only plan to use Delinea Network Information Service (adnisd) to publish network information, such as automount mount points, netgroup membership rules, or custom maps, you can skip this chapter.

Deciding to Use Agentless Authentication

Normally, the adclient agent is installed locally on a computer to handle all account authentication and lookup requests that need to be passed to Active Directory. On computers and devices where you cannot install a Delinea Agent locally, you may be able to use the Delinea Network Information Service (adnisd) to provide agentless authentication.

With agentless authentication, computers that have older or unsupported operating systems that can be, or already are, configured as NIS clients can submit NIS requests to the Delinea Network Information Service. The Delinea Network Information Service can then check its cached Active Directory information to verify whether a user or group has valid credentials and is authorized to log on.

The following figure provides a simplified view of this environment.



In this scenario, the Delinea zone acts as the NIS domain for a group of computers or devices that are configured as NIS clients. Those clients submit requests to the Delinea Network Information Service, adnisd, listening on the NIS port.

The Delinea Network Information Service periodically contacts the adclient agent to get updated information from Active Directory and generates a set of "maps" that it stores locally. The Delinea Network Information Service can then use the information in these maps to respond to NIS client requests for authentication or other services.

The user and group "maps" are generated automatically based on the users and groups that have profiles currently enabled in the zone. Network information and custom maps can also be published for a zone, but those maps must be manually imported or created. The maps for agentless authentication only require you to add and enable a profile for each Active Directory user and group who should have access to the zone. In this way, the Delinea Network Information Service can be used to service agentless authentication requests from computers or devices where adclient itself cannot be installed.

Planning for Agentless Authentication

In planning a deployment that supports agentless authentication for NIS clients, you should keep in mind that the zone associated with the computer where adnisd is installed defines the scope of information available to the NIS clients that the adnisd process serves. Each instance of adnisd supports one and only one zone, which is equivalent to a single NIS domain. The adnisd process can only look up information for the computers, groups, and users that exist in the same zone as the local computer account, and all instances of the adnisd in the same zone respond to queries using the same information from Active Directory.

For users and groups to be available for agentless authentication, therefore, they must be enabled for the zone the Delinea Network Information Service serves. In addition, each zone that supports agentless authentication requires an Active Directory attribute for storing the password hash for UNIX-enabled

users. The password hash is not created in Active Directory by default, so it must be generated then maintained using a password synchronization service installed on all of your domain controllers. The Active Directory attribute that holds the password hash must be accessible to the computers you are using as NIS servers in each zone.

> **Note**: If you install the Delinea Network Information Service on multiple computers, whether in the same zone or across multiple zones, all of these instances are zone-specific peers. There are no master/slave instances.

If you decide you want to use the Delinea Network Information Service to support agentless authentication, you should:

- Identify the zones for which you want to publish information. For example, if you want user and group information broadly available to NIS clients across the network and you have a parent zone, you may want to allow agentless authentication for all of the users in that zone. If you want to strictly control which users can be authenticated to NIS clients, you may want to create a separate agentless-authentication child zone that only contains those users and their groups. For each zone that supports agentless authentication, you must specify the Active Directory attribute for storing the password hash.
- Identify the computers that should service NIS client requests in each zone. You can designate any computer that has the Delinea Agent installed to also act as the Delinea Network Information Server in the zone. Any computer you want to use as the NIS server must have the Delinea Agent for *NIX installed and must be joined to an Active Directory domain.
- Install a password synchronization service on all of the domain controllers in the joined domain.
- Install and configure the Delinea Network Information Service (adnisd) on the selected computers in each zone. On the computers that will act as NIS servers in a zone, you must manually install and start the adnisd service. Alternatively, you can modify the startup script on each local computer so that the adnisd process starts whenever the local computer is rebooted. You also may want to customize the configuration parameters that control the operation of the adnisd process.
- Configure computers and devices as NIS clients that bind to the Delinea Network Information Service on the selected computers in each zone. If any existing NIS servers are running, you will need to reconfigure the NIS clients on the network to use the computer where the Delinea Network Information Service is installed as their NIS server.
- Import and enable the users and groups who need to be authenticated to NIS clients for the zone. You can migrate this information from existing NIS servers or local configuration files by importing passwd and group NIS maps or local /etc/passwd and /etc/group files using the **Import from Unix** wizard, or you can manually or programmatically create UNIX profiles for users and groups, as needed. The users and groups must have UNIX profiles stored in Active Directory and enabled for the local computer's zone for the Delinea Network Information Service to generate the maps it needs to service agentless authentication and lookup requests from NIS clients.
- Import and manage any additional NIS maps you want to make available to NIS clients. For example, you can import network maps such as netgroup and automount NIS maps or create custom maps using the Access Manager console.

  > **Note**: Importing existing NIS maps simply provides a mechanism for migrating information to the Active Directory. Once the information is stored in Active Directory, any original NIS maps you imported are no longer used. Instead, the Delinea Network Information Service uses the information stored in Active Directory to automatically generate the maps it needs to service authentication and lookup requests. This local cache of data is updated at a regular interval.

Selecting a Zone to Use for NIS Authentication

A computer's zone is equivalent to a NIS domain for the Delinea Network Information Service. Each instance of the Delinea Network Information Service supports one and only one zone. All instances of the Delinea Network Information Service in the same zone respond to queries using the same information from Active Directory.

If user information from a zone needs to be available to NIS clients for agentless authentication, the Delinea Network Information Service must be able to access the password hash for zone users. However, because Active Directory does not generate a password hash for users by default, there's no default attribute for storing this information.

To enable the password hash to be stored for users in a zone:

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the zone that will service NIS client requests, right-click, then click **Properties**.

   For example, if you want to work with a child zone, sanfrancisco, expand the parent zone and Child Zones nodes, select the sanfrancisco zone right-click, then click **Properties**.

4. On the General tab, select the **Support agentless client** option.
5. Select the Active Directory attribute to use for storing the password hash.

   Depending on the password synchronization service you are using and the Active Directory schema, select one of these attributes:

   - **altSecurityIdentities** if you are using the Delinea Password Synchronization program. Do not select this option if you are using a Microsoft password synchronization service.
   - **msSFU30Password** if you are using the Microsoft Windows Services for UNIX Password Synchronization Service. If you are using the Delinea Password Synchronization program, you can choose this attribute if you have the SFU schema installed.
   - **unixUserPassword** if you are using the Microsoft UNIX Identity Management Service and are using the Delinea Password Synchronization program.

6. Verify the default NIS domain name.

   By default, the zone name is used as the NIS domain name because this makes it easy to identify the scope of the information available to NIS clients. You can type a different name in the zone properties if you choose. Whether you use the default name or another name for the NIS domain, you must use the same name when you configure the NIS clients. For more information about configuring NIS clients, see Configuring NIS clients.

7. Click **OK** to save the changes and close the zone Properties.

Selecting a Computer for NIS Authentication

You can designate any computer in a zone to act as the NIS server for the zone by setting the **Allow this computer to authenticate NIS users** computer property as described in "Adding Support for Agentless Clients" in the *Administrator's Guide for Linux and UNIX*. For example, expand the Computers node in the zone that will service NIS client requests, select the computer account, right-click to select **Properties**, then click the **Delinea Profile** tab to set this option.

The computer account acting as a NIS server for the zone must be able to access the attribute containing the password hash for agentless authentication to be successful.

Selecting **Allow this computer to authenticate NIS users** adds the computer account to the`zone_nis_servers` Active Directory group. Computer accounts that are placed in the `zone_nis_servers` group are automatically granted permission to read the attribute that stores the password hash for users in the zone.

This property setting enables the computer account to access the password hash so that it can authenticate users in response to NIS client requests. However, you must manually install and start the Delinea Network Information Service on the physical computer before the computer can act as a NIS server.

Configuring a Password Synchronization Service

The Delinea Network Information Service must be able to retrieve the current password hash for zone users in order for it to respond to agentless authentication requests from NIS clients. Active Directory, however, does not generate a password hash for users by default. This task is handled by the password synchronization service.Therefore, to generate the password hash for zone users, you first need to install a password synchronization service.

You can install the password synchronization service with the Server Suite or separately using a standalone setup program. Once deployed, it ensures the passwords served by the Delinea Network Information Service are always up-to-date. With a password synchronization service, any time users change their Active Directory password, the corresponding password hash in their user profile is updated to reflect the change. Depending on your environment, you can choose to install one of the following:

- Delinea Password Synchronization program
- Microsoft Windows Services for UNIX Password Synchronization Service
- Microsoft Windows UNIX Identity Management Service

Note: Regardless of the password synchronization service you choose to use, the service must be installed on all domain controllers in the Active Directory domain where you are enabling agentless authentication.

Using Delinea Password Synchronization

You can install the Delinea Password Synchronization program using the Server Suite setup program. Alternatively, you can install Delinea Password Synchronization independent of the the Server Suite using it own setup program. If you install the Delinea Password Synchronization program using the setup program, you can skip this section.

To install the Delinea Password Synchronization program:

1. Copy the CentrifyDC_PasswordSync-n.n.n-win64 package to your Active Directory domain controller.
2. Open the CentrifyDC_PasswordSync-n.n.n-win64 executable or Microsoft software installation (.msi) file to start the setup program.

   Note: You can run the setup program interactively or silently if you use the Microsoft software installation (.msi) file. If you are installing silently using the msiexec program, you can skip the steps in this section.
3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement. If you accept the license agreement, select **I accept the terms of the license agreement**, then click **Next**.
5. Type your name and company, select who should be able to use this application on the computer, then click **Next**.
6. Select a restart option, then click **Finish**.

Once installed, the Delinea Password Synchronization program will generate the initial password hash when users next change their password, then update the password hash at each password change thereafter. The password hashes are created using DES encryption with a two character salt. If the password hash is stored in the altSecurityIdentities attribute, it has a prefix of cdcPasswordHash, for example:

```
cdcPasswordHash:VkievQ69VhYKc
```

If the password hash is stored in one of the other supported attributes, it is stored without a prefix.

When a user changes his Active Directory password, the Delinea Password Synchronization program discovers the zones to which that user has access and updates the appropriate attribute that holds the password hash for that user in each zone.

> **Note**: The initial password hash is only generated when the user changes his password. You may want to force users to change their password at the next logon to get the password set at the earliest opportunity. Client authentication requests may fail for users who do not have a password hash available. If the password hash field in the passwd.byname or passwd.byuid map displays a single exclamation point (!), it indicates that the user's password hash has not been set.

Using Microsoft Password Synchronization Service

If you choose to use one of the password synchronization services provided by Microsoft instead of the Delinea Password Synchronization program, follow the instructions provided with the software to install the service. In general, you need to do the following to use the Microsoft password synchronization services:

- Set the Windows domain to the domain you joined after installing the Delinea Agent for *NIX.
- Set the NIS domain name to the zone name you specified when you joined the domain. For example, if you are using the **default** zone, set the NIS domain to **default**. Although you can set the NIS domain name to something other than the zone name when creating or modifying a zone's properties, you must use the zone name for this setting if you use Microsoft password synchronization.
- Set the NIS Server name to the host name of the computer running both the adclient and adnisd services.
- Give user accounts access to the zone and NIS domain. If you are using the Microsoft Windows Services for UNIX, select the zone name from the list of NIS domains on the **UNIX Attributes** tab.

  The rest of the fields on the UNIX Attributes tab are not used by Server Suite, but you are required to enter information for these fields to enable the NIS domain for the user. Therefore, you should specify a UID, Login shell, Home directory, and Primary group for the user account, then click **OK**.

Locating Zones for Password Synchronization

Only Active Directory users with a UNIX profile created using the Access Manager console include the attribute (parentLink) needed to look up their zone information for password synchronization. You can use the **Orphan Unix data objects** option in the Analyze Wizard to check the forest for accounts missing this attribute setting and attempt to correct the problem.

If the Analysis Results display a **Warning** for the **Orphan Unix data objects** check, you can right-click, then select **Properties** to view additional details. If the profile is missing the parentLink attribute, select the warning, right-click, then select **Populate parentLink** attribute to define this attribute for the user.

For more information about troubleshooting issues for the Delinea Network Information Service, see Troubleshooting and Logging NIS Operations. For more information about using the Analyze wizard in the Access Manager console, see "Analyzing information in Active Directory" in the *Administrator's Guide for Linux and UNIX*.

This section describes how to install and configure the Centrify Network Information Service (adnisd). The adnisd process allows a Centrify-managed computer to act as the NIS server for NIS clients in a joined domain. Using adclient and adnisd together, you can store authentication, authorization and network information in Active Directory, and respond to NIS client requests from computers and devices even where adclient cannot be installed.

Installing the Centrify NIS Server

Whether you want to use the Centrify Network Information Service for agentless authentication, managing network information, or publishing custom maps, you must install and configure adnisd on at least one computer in at least one zone before you can begin responding to NIS client requests.

In most cases, adnisd is installed as part of a custom installation of the Server Suite or as a separate software package, independent of the installation of adclient. The naming convention for the standalone software package is:

```
centrifydc-nis-n.n.n-os-architecture
```

Keep in mind:

- You must install adnisd on a computer where adclient is also installed.
- The Active Directory domain and zone the local computer has joined defines the NIS domain, and therefore the information available to NIS clients.
- You cannot use adnisd to serve NIS maps if your managed computer joined the domain using the --workstation option.
- Using the --workstation option adds a computer to the single Auto Zone where user and group profiles are generated automatically. Computers in the Auto Zone cannot be used as NIS servers or NIS clients.
- You can install adnisd using any installation program appropriate for the local operating environment, such as RPM, SMIT or YAST.
- If you are upgrading from a previous release of Server Suite and have an earlier version of adnisd, stop the existing adnisd service and use install.sh to remove the old packages before installing the new version of adclient and adnisd.

The following steps are only an example of how to install adnisd locally on a computer. The specific steps required depend on the local operating environment and the installation program you choose.

1. As root on the managed computer, use adinfo to verify that adclient is installed, and that the local computer is joined to a domain and can connect to Active Directory:

```
su -
Password:

adinfo

Local host name:     magnolia
Joined to domain:    ajax.org
Joined as:           magnolia.ajax.org
Current DC:          ginger.ajax.org
Preferred site:      Default-First-Site-Name
Zone:                ajax.org/Program Data/Centrify/Zones/default
Last password set:   2006-12-28 14:47:57 PST
CentrifyDC mode:     connected
```

2. Copy the package appropriate to the local computer's operating environment, from the Server Suite CD or a download directory, to a local directory.

   For example, if the operating environment is Solaris 9 SPARC:

   ```
   cp /tmp/centrifydc-nis-n.n.n-sol8-sparc-local.tgz .
   ```

3. If the package is a compressed file, unzip and extract its contents. For example, on Solaris:

```
gunzip -d centrifydc-nis-n.n.n-sol8-local.tgz
tar -xf centrifydc-nis-n.n.n-sol8-sparc-local.tar
```

4. Run the appropriate command for installing the package. For example, on Solaris:

   ```
   pkgadd –d CentrifyDC-nis -a admin
   ```

Adding IP Addresses from Which to Accept Requests

By default, the Network Information Service accepts only local NIS client requests. To accept requests from any other NIS clients in a network, modify nisd.securenets in the /etc/centrifydc/centrifydc.conf file to specify the computer subnets from which to accept NIS requests. This parameter configures adnisd to filter NIS client requests by IP address. It ignores all other NIS client requests.

For example, to restrict NIS requests to the single trusted subnet 172.68.0.0, add a line similar to the following to nisd.securenets:

```
nisd.securenets: 172.68.0.0/255.255.0.0
```

To specify multiple subnets, separate the entries with commas or spaces:

```
nisd.securenets: 172.68.0.0/255.255.0.0,196.48.0.0/0
```

To accept NIS client requests from any computer, use this:

```
nisd.securenets: 0/0
```

On systems with multiple Ethernet interfaces, adnisd configures RPC to the first interface. If an NIS client is trying to communicate on a different interface, adnisd will not receive the request.

Before creating sockets, adnisd reads the centrifydc.conf file to see if an IP address and TCP and UPD ports are specified. If not, it uses localhost and random port numbers assigned by the operating system.

You set the IP address, TCP port and UDP port using the nisd.net_addr, nisd.port.tcp, and nisd.port.udp configuration parameters, respectively in the centrifydc.conf file.

For more information, see the *Configuration and Tuning Reference Guide*.

Starting the adnisd Process

After you have specified the subnets from which to accept NIS client requests, you can either manually start the adnisd process at the command line, or reboot the local computer. By default, the adnisd process starts automatically whenever the computer is rebooted. If you don't want the process started automatically, you should modify the startup script on the local computer to remove adnisd from the processes started.

> **Note**:: The installer adds the adnisd process to a computer's startup script for you. If you are not importing NIS maps right away, you may want to modify the startup script to prevent the adnisd process from starting before you are ready to begin servicing client requests.

To start the adnisd process at the command line:

1. Verify that adclient is running and the local computer is joined to a domain.
2. Verify that RPC is running on the local computer. For example:

   rpcinfo -p localhost

   The adnisd process requires RPC services. If you restart RPC, you also need to restart the adnisd process.
3. Type the appropriate start command. For example, on Red Hat Linux, type:

   ```
   /sbin/service adnisd start
   ```

   On most other platforms, run:

   ```
   /etc/init.d/adnisd start
   ```

   On Solaris 10 or later, the daemon is controlled through the Solaris Service Management Facility. For example:

   svcadm enable nis/centrifydc-server

When the adnisd process starts, it connects to Active Directory through adclient and does the following:

- Retrieves the current user, group, network and custom information stored in Active Directory for its zone.
- Generates additional maps derived from the retrieved information, such as netgroup.byuser, netgroup.byhost, passwd.byuid, passwd.byname, group.byname, and group.bygid.
- Stores the information retrieved or derived from Active Directory in a local cache of NIS map data.

After the initial connection, adnisd periodically connects to Active Directory through adclient to retrieve updated information for its zone. However, adnisd always responds to NIS client requests using the data in its local cache so that it can respond to NIS requests even if Active Directory is unavailable.

Customizing the Update Interval for NIS Maps

By default, every 30 minutes (1800 seconds), adnisd uses adclient to connect to Active Directory. At the update interval, adnisd does the following:

- Checks for network NIS maps explicitly defined in Active Directory to determine whether any records have changed.
- Generates derived maps for any explicitly defined network maps that adnisd recognizes. For example, if the netgroup map is found in Active Directory, adnisd generates the netgroup.byuser and netgroup.byhost maps.
- Updates the local cache with all changes to the network NIS maps.
- Updates the local cache with changes to the derived maps for user and group information in the zone.

Note: In most cases, updating the local cache of NIS data does not require you to restart any services.

In most organizations, the default update interval is adequate. In a more volatile or stable NIS map environment, reduce or increase the time between updates, as appropriate, by modifying the nisd.update.interval parameter in /etc/centrifydc/centrifydc.conf to specify a different number of seconds between updates; for example:

```
nisd.update.interval: 900
```

For more information, see the *Configuration and Tuning Reference Guide*.

Customizing the NIS Maps to PUblish

By default, the adnisd process retrieves all NIS maps stored in Active Directory at each update interval, updates its local cache as needed, and makes all such data available to its NIS clients. In some cases, you may want to prevent NIS clients from accessing data in specific maps or from looking up information using a specific key.

If you want to customize the list of maps to make available to NIS clients, modify the nisd.maps or nisd.exclude.maps parameter in /etc/centrifydc/centrifydc.conf, or apply a group policy.

- With the nisd.maps parameter, you explicitly list the NIS maps, including derived maps, to *include* in the local cache of map data; for example:

  ```
  nisd.maps: hosts.byname,hosts.byaddr,automount
  ```

- With the nisd.exclude.maps parameter, you list the NIS maps to *exclude* from responses to NIS client requests (typically user and group information). When you specify a map, its derived maps are excluded as well. For example:

  ```
  nisd.exclude.maps: group passwd
  ```

For more information, see the *Configuration and Tuning Reference Guide*.

Configuring the Maximum Number of Map Sets

When adnisd receives data for explicitly-defined NIS maps, the data comes from the domain controller selected by the adclient process. If the domain controller the adclient process has changed – for example, if it is unavailable – the adclient process attempts to find another available domain controller.

To ensure the data consistency of the NIS maps retrieved from Active Directory, adnisd keeps a separate set of NIS records from each domain controller. This enables adnisd to switch between domain controllers efficiently, but uses more space in the local cache.

You can control the maximum number of alternate sets of NIS maps to maintain (default is two) by modifying the nisd.maps.max parameter in /etc/centrifydc/centrifydc.conf. For example, to keep up to four sets of NIS maps, specify:

```
nisd.maps.max: 4
```

For more information, see the *Configuration and Tuning Reference Guide*.

Handling Large Active Directory Groups

In most cases, the NIS server cannot send more than 1024 characters of data to NIS clients in response to a query. This limitation can create problems when the NIS client requests information for a large group with a long membership list. By default, the adnisd process automatically truncates the list at 1024 characters.

You can configure adnisd to split large groups into several groups of conforming size and names using nisd.largegroup.suffix and nisd.largegroup.name.length in /etc/centrifydc/centrifydc.conf.

Splitting a single large group into multiple new groups

If you specify any value for the nisd.largegroup.suffix parameter,adnisd splits large groups into multiple new groups automatically, creating a new group whenever a group's data size exceeds 1024-character limit by appending the string you define in nisd.largegroup.suffix plus a sequential number.

For example, if you have a large group named performix-worldwide-corp, and have defined the suffix string as "-all", when the performix-worldwide-corp group membership is split into multiple groups, the groups are named as follows:

```
performix-worldwide-corp-all1
. . .
performix-worldwide-corp-all n
```

All of the new groups have the same group identifier (GID) as the original group.

Setting the maximum length of new group names

If the new group names would exceed the maximum length for group names on a platform, use the nisd.largegroup.name.length parameter. If you do this,adnisd` truncates the original group name so as not to exceed the maximum name length.

For the example above, if you set a maximum name length of 14, the split groups are named:

```
performix-all1
...
performi-all10
...
perform-all100
```

All of the new groups have the same group identifier (GID) as the original group.

For more information, see the *Configuration and Tuning Reference Guide*.

Making the NIS Server Available

After you install and configureadnisd on a computer, you must configure other computers or devices on the network to use the computer runningadnisd for NIS client requests.

In general, configuring NIS clients to use the Centrify Network Information Service involves:

- Stopping any existing legacy NIS server processes.
- Modifying the NIS client's configuration file to identify the zone and computer name of the computer where theadnisd process is installed.
- Sending a bind request from the NIS client to the new Centrify NIS server.

For more information, see Configuring NIS clients.

This section describes how to configure NIS clients to receive authentication, authorization, and network information through the Delinea Network Information Service.

Specifying the Server for NIS Clients to Use

After you install and configure adnisd on a computer, you must configure other computers or devices to send their NIS lookup requests to the computer running adnisd. The specific steps for configuring the NIS client are slightly different in different operating environments. In general, configuring NIS clients involves:

- Stopping the connection to any existing NIS server.
- Identifying the zone and computer name of the computer where adnisd is installed in the client's NIS configuration file.
- Binding to the new Delinea NIS server.
- Restarting services that use NIS, or rebooting the computer.

For information about configuring the NIS client in different operating environments, see the appropriate section below.

> **Note**: The client configuration instructions assume that you are using the zone name as the NIS domain name. If not, substitute the NIS domain name you specified when you created the zone where applicable. For more information about configuring NIS clients on any specific platform and OS version, consult the documentation for that platform.

Configuring NIS Clients on Linux

To configure the NIS client on a Linux computer:

1. Stop any running NIS service and remove all files from the /var/yp/binding directory. For example, run the following commands:

   ```
   /sbin/service ypbind stop
   rm -rf /var/yp/binding/*
   ```

2. Set the NIS domain name for the client to the zone name or NIS domain name of the computer where the adnisd process is running.

   ```
   domainname zone_name
   ```

   For example, if you have installed adnisd on a computer in the corpHQ zone:

   ```
   domainname corpHQ
   ```

3. Edit the NIS configuration file, /etc/yp.conf, to specify the Delinea zone and the name of the computer where adnisd is installed.

   ```
   domain zonename server hostname
   ```

   For example, add a line similar to this to /etc/yp.conf:

   ```
   domain corpHQ server localhost
   ```

   If your NIS clients are configured for broadcast discovery, this step may not be necessary.

4. Start the ypbind service.

   On Red Hat Linux, run:

   ```
   /sbin/service ypbind start
   ```

   On Debian 3.1, run the nis script (controlled using the file /etc/default/nis). By default, the script starts the NIS client, ypbind. For example, run the following command:

   ```
   /etc/init.d/nis start
   ```

   One SuSE Linux 9.3 Professional, run:

   ```
   /etc/init.d/ypbind start
   ```

5. Modify the passwd, group, and shadow lines in /etc/nsswitch.conf file to use compat as the source:

   ```
   passwd: compat
   group: compat
   shadow: compat
   ```

6. Restart services that rely on the NIS domain, or reboot the computer to restart all services. The most common services to restart are autofs, NSCD, cron and sendmail.

Configuring NIS Clients on Solaris

To configure the NIS client on a Solaris computer:

1. Stop any running NIS service and remove all files from the /var/yp/binding directory. For example, run the following commands on Solaris 8 or 9:

   ```
   kill ypbind
   rm -rf /var/yp/binding/*
   ```

   On Solaris 10, stop the service by running:

   ```
   svcadm disable network/nis/client
   ```

2. Set the NIS domain name for the client to the zone name of the computer where adnisd is running.

   ```
   domainname zone_name
   ```

   For example, if you have installed adnisd on a computer in the corpHQ zone:

   ```
   domainname corpHQ
   ```

3. Run the ypinit -c command and enter the name of the computer where adnisd is installed.

This step is not required if you use the broadcast option to locate the server when you run the ypbind command. You must use ypinit, however, if your network topology would prevent a broadcast from reaching the desired servers. For example, if the router does not transmit broadcasts across subnets, use the ypinit -c command to specify a server on a different subnet.

Start the ypbind service. On most versions of Solaris, run:

```
/usr/lib/netsvc/yp/ypbind
```

If you are using the broadcast option to locate the server, start the service with that option. For example:

```
/usr/lib/netsvc/yp/ypbind -broadcast
```

On Solaris 10, run:

```
svcadm enable network/nis/client
```

Modify the passwd, group, and shadow lines in /etc/nsswitch.conf file to use compat as the source:

```
passwd: compat
group: compat
shadow: compat
```

Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are autofs, NSCD, cron and sendmail.

Configuring NIS Clients on HP-UX

To configure the NIS client on an HP-UX computer:

1. Stop any running NIS service and remove all files in the /var/yp/binding directory. For example, run the following commands:

   ```
   /sbin/init.d/nis.client stop
   rm -rf /var/yp/binding/*
   ```

2. Edit the NIS configuration file, /etc/rc.config.d/namesvrs, to set the NIS_CLIENT to 1 and the NIS_DOMAIN to the name of the Delinea zone. For example:

   ```
   NIS_CLIENT=1
   NIS_DOMAIN="zone-name"
   ```

3. Add the -ypset option to the YPBIND_OPTIONS variable and set the YPSET_ADDR variable to the IP address of the computer where adnisd is installed. For example:

   ```
   YPBIND_OPTIONS="-ypset"
   YPSET_ADDR="15.13.115.168"
   ```

This step is not required if you want to use the broadcast option to locate the server when you run the ypbind command.

4. Set the NIS domain name for the client to the zone name of the computer where theadnisd process is running.

   domainname zone_name

5. Start the ypbind service. On HP-UX, you can start the service by running:

   ```
   /sbin/init.d/nis.client start
   ```

6. Modify the passwd, group, and shadow lines in /etc/nsswitch.conf file to use compat as the source:

   ```
   passwd: compat
   group: compat
   shadow: compat
   ```

7. Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are autofs, pwgrd, cron and sendmail.

Configuring NIS Clients on AIX

To configure the NIS client on an AIX computer:

1. Stop any running NIS service and remove all files from the /var/yp/binding directory. For example, run:

   ```
   stopsrc –s ypbind
   ```

   If the computer is not already a NIS client, you can use the System Management Interface Tool (smit) and the mkclient command to add adnisd to the computer.

2. Open the /etc/rc.nfs file and verify that the startsrc command is configured to start the ypbind daemon:

   ```
   if [ -x /usr/etc/ypbind ]; then
    startsrc -s ypbind
   fi
   ```

3. Set the client's NIS domain name to the zone name of the computer whereadnisd is running. For example:

   ```
   domainname zone_name
   ```

4. Start the ypbind service:

   ```
   startsrc -s ypbind
   ```

5. Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are autofs,NSCD, cron and sendmail.

   Note: Theadnisd service is not supported in a workload partitioning (WPAR) environment (Ref: CS-30588c).

Verifying the Client Configuration

Run the domainname command to verify that the client is configured to use the appropriate Delinea zone or NIS domain name. For example, if you have configured a computer to service NIS requests for the sanfrancisco zone and are using the zone name as the NIS domain name:

```
domainname
sanfrancisco
```

To test that the client can connect to the Delinea Network Information Service, run one or more NIS client request commands; for example:

```
ypwhich
ypwhich -m
ypcat -k mapname
```

Checking the Derived passwd and Group Maps

On a computer you have configured as an NIS client, verify that the NIS maps required for agentless authentication are available by running the following command:

```
ypwhich -m
```

At a minimum, you should see the passwd.* and group.* map names, followed by the name of the computer you are using as the NIS server. For example, if the computer running adclient andadnisd is iceberg-hpux, you should see output similar to this:

```
passwd.byuid iceberg-hpux
passwd.byname iceberg-hpux
group.byname iceberg-hpux
group.bygid iceberg-hpux
```

These passwd.* and group.* maps are automatically generated based on the information stored in Active Directory for the zone, including all Active Directory users and groups granted access to the zone. You can view information from any of these maps using a command like ypcat passwd.byname. The output displayed should look similar this:

```
paul:Xq2UvSkNngA:10000:10000:paul:/home/paul:/bin/bash
mlopez:!:10002:10000:Marco Lopez:/home/mlopez:/bin/bash
jsmith:!:10001:10000:John Smith:/home/jsmith:/bin/bash
```

In this example, the user paul has a password hash, but users mlopez and jsmith do not.

If a user account is new, disabled, locked, requires a password change, or is not enabled for a zone, the Delinea NIS server sets the user's hash field to "!"

> **Note**: On some platforms, you may see ABCD!efgh12345$67890 as the password hash for users who need to set their password.

This section describes how to import, create and manage NIS maps and map entries using the Access Manager console.

> **Note**: You can also use ADSI Edit, ADEdit, custom scripts or other tools to add, modify and remove NIS maps and map entries. To import NIS maps, however, you must use the Access Manager console.

Importing and Creating User and Group Profiles

If you want to make user and group information available to NIS clients, whether for agentless authentication or in response to other lookup requests, you must first make sure the appropriate users and groups have zone profiles and role assignments defined in the zone. The zone information is used for automatic generation of the maps passwd.byname, passwd.byuid, group.byname, and group.bygid. If you disable a user profile in the zone, the user's information cannot be retrieved or published in response to NIS client requests, or used to authenticate the user's identity.

You can import existing user and group information directly from existing NIS servers and domains or from properly formatted text files, such as local /etc/passwd and /etc/group files, using the **Import from UNIX** wizard, or you can create new profiles for Active Directory users using the Access Manager console.

Once the appropriate user and group profiles have been added to the zone you are using as a NIS domain, the information is available to NIS clients unless you explicitly restrict the publication of this information.

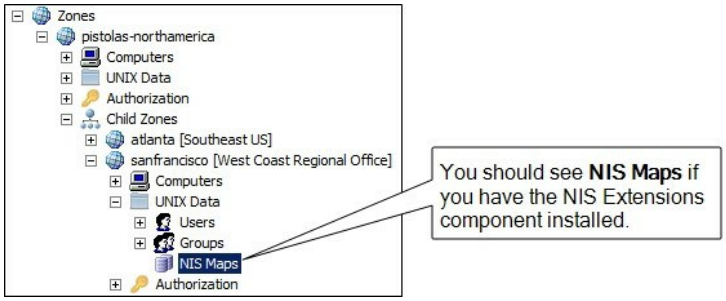
> **Note**: For information about restricting the maps published, see Customizing the NIS maps to publish. For information about importing or creating user and group profiles in a zone, see the *Administrator's Guide for Linux and UNIX*.

Publishing Network or Custom Information

In addition to user and group information, adnisd can publish network information or make custom information available to NIS clients. For example, you can import information from standard NIS maps such as automount, netgroup, and automaster, if these maps exist in your environment. Importing network information or creating custom maps, however, requires you to have the NIS Extensions.

> **Note**: NIS Extensions are installed by default when you run the setup program. If you did not select this option, rerun the setup program and select **NIS Extensions** from the list of Access Manager Administration components.

If you have the NIS Extensions installed, you should see the NIS Maps node under each zone. For example, if you are using hierarchical zones, you can see NIS Maps under the UNIX Data node for the parent or child zone you select:



Importing Network NIS Maps

To use Access Manager to import a standard network NIS map into Active Directory:

1. Open Access Manager.
2. In the console tree, navigate to the specific zone into which you want to import NIS maps.
3. Expand the console tree to display NIS Maps.
4. Select NIS Maps, right-click, then click **Import Maps**.
5. Select whether you want to connect to the NIS server and domain or import the information from a text file, then click **Next**.

- If you are importing maps directly from an existing NIS server, type the name of the **NIS domain** and **NIS server**. Using this option requires network connectivity to the NIS server from the Window computer you are using.
- If you are importing a map from a text file, click **Browse** to navigate to the map file you want to import. If you cannot connect directly to the NIS server, you should export the NIS database to a file; then import the information using this option.

6. Select the NIS maps to import if you are importing directly from an existing NIS server, or type a map name and define the file format if importing from a file, then click **Next**. The Import Maps wizard does not validate the information to be imported. If the map has invalid entries, they are imported as-is.

   If you importing from a text file:

   - Type a **Map name** that describes the type of map being imported. In most cases, you should use the base name that identifies the configuration file used to generate the NIS database. For example, use hosts to identify the map generated from the /etc/hosts file.
   - Type the **Field separator** character used to separate fields in the map file.
   - Type the column number that defines the start of the **Key field**.
   - Specify any additional options as appropriate for the file you are importing. For example, select **Comments are included in the file after** and type the character used to designate comments if the file includes comments.

     For Access Manager to correctly interpret the map file, you need to provide accurate information about the file format, such as the type of separator used between fields.

     Because the Centrify NIS server does not include comments in response to service requests, you must save the map to a text file and import from that file to retrieve comments contained in NIS maps.

7. When the import is complete, click **Finish**.
8. After importing NIS maps, restart the adnisd service.

Creating New NIS Maps in Active Directory

If you cannot import network information from existing NIS maps, you can create new maps by adding the appropriate information directly to Active Directory using Access Manager. Once you add the information to Active Directory, adnisd can use the information to automatically generate a local cache of the map data and make the information in those generated maps available to NIS clients.

> **Note**: If you are creating NIS maps manually, keep in mind that the Network Information Service can return a maximum of 1024 characters of data in response to a query from any NIS map, so make sure all NIS map entries have less than 1024 characters of data.

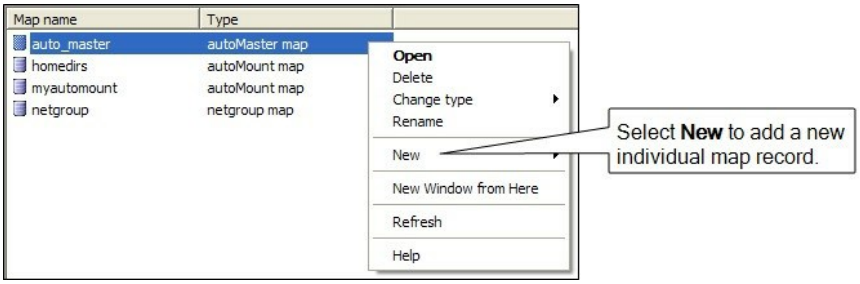
To create a new network NIS map in Active Directory

1. Open Access Manager.
2. Navigate to the specific zone for which you want to create maps.
3. Expand the console tree to display NIS Maps.
4. Select NIS Maps, right-click, then click **New** and select the type of map you want to create.

   For most map types, you can only use the recognized map name for the new map. Recognized map names enable you to use derived maps to retrieve information using different keys. If you are creating a new **Automaster** map, you must choose either auto_master or auto.master as the map name to retrieve the names of the automount maps.

   If you select the **Generic Map** option, you can create a custom NIS map for any key/value pairs that you want to make available to NIS clients. For more information, see Creating generic custom maps.

5. Select the new empty map, right-click, then click **New > Map Entry** or **New > netgroup** to add a new individual map record.



The file format and the specific fields used in individual map records depend on the type of map you are working with.

6. Type the appropriate information for the fields listed, then click **OK** to save a record in the new map.

   For more information about the fields required in any NIS map, see the man page for the type of map you are creating. For example, see the man page for netgroupto see detailed information about required and optional fields and the format of netgroup maps.

   You can use Active Directory groups in netgroup records. Using Active Directory groups in netgrouprecords enables dynamic changes to user and computer pairings based on their Active Directory group membership. If you have existing processes for adding and removing users and computers in Active Directory groups, you can leverage those processes in netgroup records.

Creating Maps for Common Network Services

Centrify uses explicitly-defined NIS maps to generate derived maps automatically. Once a recognized base map is imported or created manually in Active Directory, the agent generates and stores its derived maps so that information can be retrieved searching on different keys.

> **Note**: In most cases, you can import recognized base maps directly from an existing NIS server and domain or from generated text files (for example, files created using the niscat command). Alternatively, you can create the base maps manually using the corresponding map type in Access Manager.

The following table describes the recognized base maps and their derived maps.

aliases

The aliases map is the abbreviated name for the mail.aliases map. The derived maps are mail.aliases and mail.byaddr. In most cases, the NIS map is created from the /etc/aliases or /etc/mail/aliases file. A typical line looks like this:

```
alias: address1 [address2 addressn...] # comment
```

For example:

```
acme: amy.adams@acme.com bill.byrnes@acme.com
widgetco: aaron@widgetco.com,...,zuza@widgetco.com
```

For the mail.alias map, the entries are defined like this:

- Key is the alias name: acme

- Value is the list of addresses for the alias: amy.adams@acme.com bill.byrnes@acme.com

For the mail.byaddr map, the entries are defined like this:

- Key is an address: amy.adams@acme.com
- Value is the corresponding alias: acme

If you create an aliases map in Active Directory, you must include the key as part of the value. For example:

- Key: acme
- Value: acme: someone@acme.com
- Comment: someone@acme.com is the address

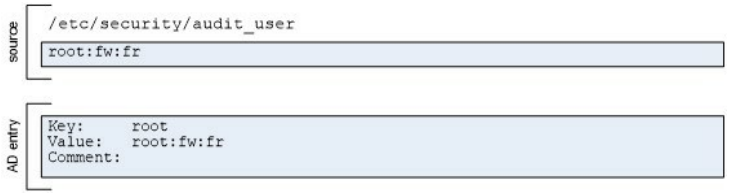
audit_user

In most cases, the audit_user map is created from the /etc/security/audit_user file. A typical line looks like this:

```
user_name:always_audit_flags:never_audit_flags
```

For example:

```
root:lo:no
wily:lo,am:io,cl
kris:lo,ex,+fc,-fr,-fa:io,cl
```



For the audit_user map, entries are defined like this:

- Key is the user name: root
- Value takes the following format: user_name:always_audit_flags:never_audit_flags

If you create an audit_user map in Active Directory, you must include the key as part of the value. For example:

- Key: root
- Value: root:lo:no

This map is only applicable for Solaris.

auth_attr

In most cases, the auth_attr map is created from the /etc/security/auth_attr file. A typical line looks like this:

```
name:res1:res2:short_description:long_description:attr
```

For example:

```
solaris.:::All Solaris Authorization::help=SolarisAuth.html
solaris.user.manage:::Manage Users::help=ManageUsers.html
```

If you create an auth_attr map in Active Directory, you must include the key as part of the value. For example:

- Key: solaris.
- Value: solaris.:::AllSolarisAuthorizations::attribute
- Comment: This map provides authorization attributes for Solaris.

This map is only applicable for Solaris.

bootparams

In most cases, the bootparams map is created from the /etc/bootparams file. A typical line looks like this:

```
client_name key=value:[key=value:...]
```

For example:

```
client root=sr04:/export/client/root domain=nyc.test
engr1 root=smoketest:/export/engr1/root rootopts=:vers=2
```

If you create a bootparams map in Active Directory, the value must consist of key and value pairs. For example:

- Key: client
- Value: root=sr04:/export/client/root domain=nyc.test
- Comment: The value consists of key=value pairs separated by colons (:).

This map is only applicable for Solaris.

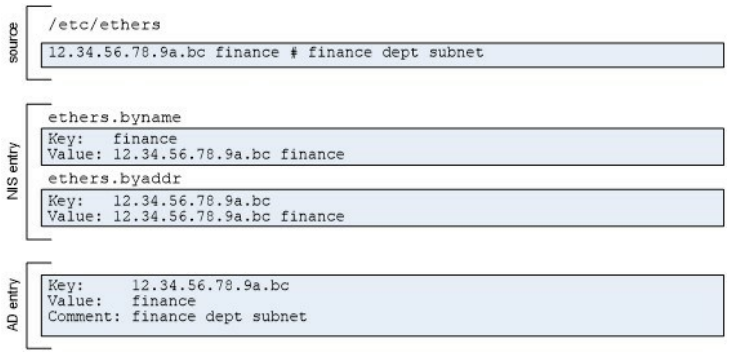
ethers

The ethers map is the abbreviated name for the ethers.by name map. The derived maps are ethers.byname and ethers.byaddr. In most cases, the NIS map is created from the file /etc/ethers file. A typical line looks like this:

```
ethernet_address host_name
```

For example:

```
52:ef:75:72:4e:c8 rhel9
31:ee:c5:72:4e:18 finance
```



For the ethers.byname map, entries are defined like this:

- Key is the host name: rhel9
- Value is the ethernet address for the host name: 52:ef:75:72:4e:c8

For the ethers.byaddr map, entries are defined like this:

- Key is an address: 52:ef:75:72:4e:c8
- Value is the host name: rhel9

If you create an ethers map in Active Directory, you must include the key as part of the value. For example:

- Key: rhel9
- Value: 52:ef:75:72:4e:c8 rhel9
- Comment: The host name for 52:ef:75:72:4e:c8 is rhel9

exec_attr

In most cases, the exec_attr map is created from the /etc/security/exec_attr file. A typical line looks like this:

```
name:policy:type:res1:res2:id:attr
```

For example:

```
Application Server Management:susuer:cmd:::/usr/bin/admin:
DBA:unix-dba:cmd:::/usr/db/bin/dbadmin:
dbuser:unix-dbuser:cmd:RO::/usr/sbin/db/opensql
```

If you create an exec_attr map in Active Directory, you must include the key as part of the value. For example:

- Key: Application Server Management
- Value: execution profile name and properties followed by attributes defined as key and value pairs for the profile:
  Application Server Management:suser:cmd::: \ /usr/appserver/bin/admin:

This map is only applicable for Solaris.

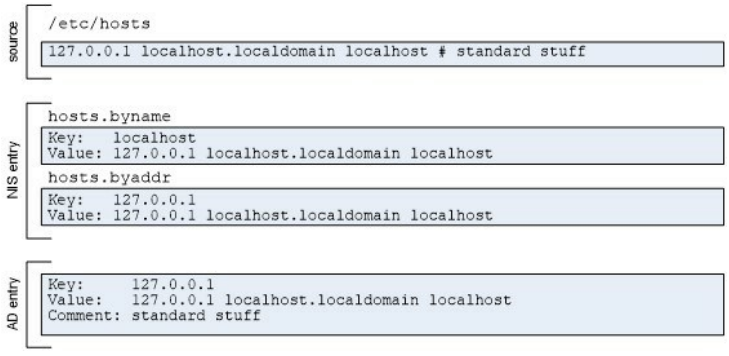
hosts

The hosts map is the the abbreviated name for the hosts.byname map. The derived maps are hosts.byname and hosts.byaddr. In most cases, the NIS map is created from the /etc/hosts file. A typical line looks like this:

```
host_ip_address host_name [alias,...] # comment
```

For example:

```
127.0.0.1 localhost.localdomain localhost
192.168.22.1 arcade.cendura.net arcade arc1 # clustername
```



For the hosts.byname map, entries are defined like this:

- Key is the host name: localhost

- Value is the IP address and any aliases defined for the host: 127.0.0.1 localhost.localdomain localhost

For the hosts.byaddr map, entries are defined like this:

- Key is an address: 127.0.0.1
- Value is the IP address and any aliases defined for the host: 127.0.0.1 localhost.localdomain localhost

If you create a hosts map in Active Directory, you must include the key as part of the value. For example:

- Key: 127.0.0.1
- Value: IP address and any aliases defined for the host: 127.0.0.1 localhost.localdomain localhost
- Comment: The value includes both the host name and IP

netgroup

The netgroup map defines a hierarchy of netgroupgroups and members. The netgroupmap controls access by user name, host name, or NIS domain name. The derived maps are netgnetgroup.byhostroup.byhost and netgroup.byuser. In most cases, the NIS map is created from the /etc/netgroup file. A typical line looks like this:

netgroup_name (host,user,NIS_domain)[,netgroup]...

The keys in a netgroupmap are the names of each netgroup. The values in a netgroupmap are one or more space-separated elements. An element can be:

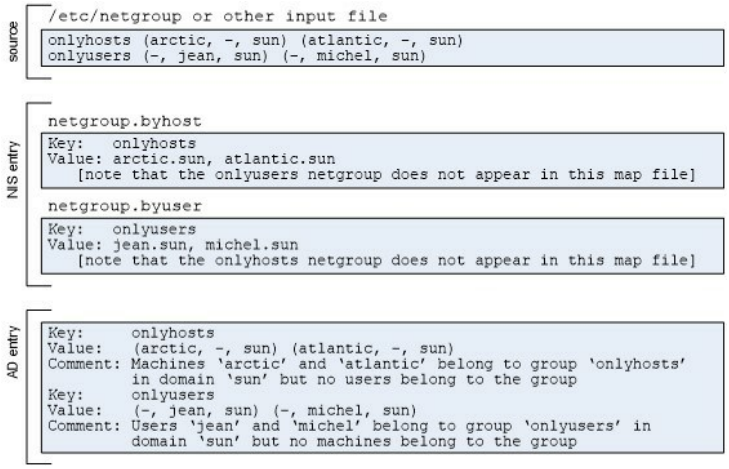
- a set of three comma-separated components.
- a netgroupname.

When specifying an element as a set of three components, you can omit any component to allow any value for that component or specify the special character dash (-) to eliminate a component as a valid value.

The netgroup.byhost map uses the host name as the key and the value is the list of all netgroups that contain the key host somewhere in the hierarchy.

The netgroup.byuser map uses the user name as the key and the value is the list of all netgroups that contain the key user somewhere in the hierarchy.

If you create a netgroupmap in Active Directory, you must not include the key as part of the value. To illustrate, the following example has entries for two netgroups—onlyhosts and onlyusers—and how the groups become key and value entries in the derived NIS maps.



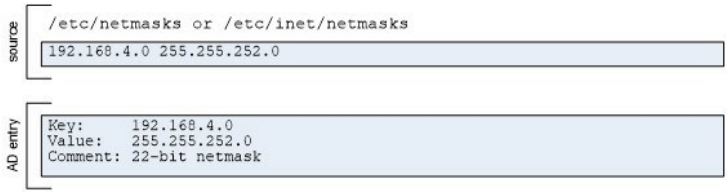
netmasks

In most cases, the netmasks map is created from the /etc/inet/netmasks or /etc/netmasks file. A typical line looks like this:

IP_addressnetmask # comment

For example

192.168.4.0 255.255.252.0
192.168.4.1 255.255.255.0



If you create a netmasks map in Active Directory, you must not include the key as part of the value. For example:

- Key: 192.168.4.0
- Value: 255.255.252.0
- Comment: This is a 22-bit netmask.

This map is only applicable for Solaris.

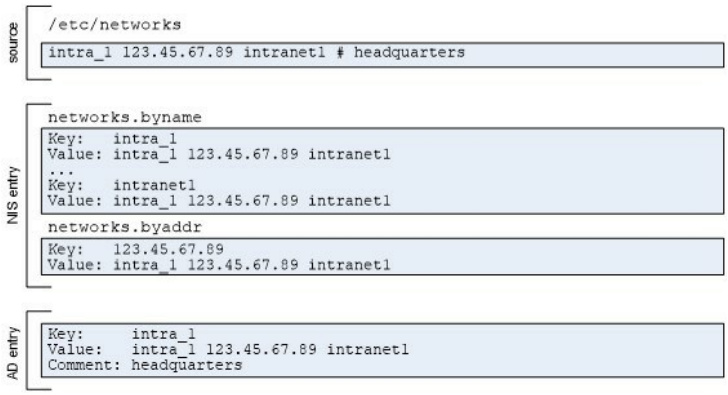
networks

The networks map is the the abbreviated name for the networks.byaddr map. The derived maps are networks.byname and networks.byaddr. In most cases, the networks map is created from the /etc/networks file. A typical line looks like this:

network_name network_address [alias1,...] # comment

For example:

```
arpa 10 arpanet
intra_1 123.45.67.89 intranet # headquarters
sf_site 171.22.0.0 sf1 # san francisco satellite
```



For the networks.byname map, entries are defined like this:

- Key is the network name: intranet
- Value is the network address and any aliases defined for the network: intranet 171.22.0.0 intra

For the networks.byaddr map, entries are defined like this:

- Key is the network address: 171.22.0.0
- Value is the network name and any aliases defined for the network: intranet 171.22.0.0 intra

If you create a networks map in Active Directory, you must include the key as part of the value. For example:

- Key: intranet
- Value: intranet 171.22.0.0 intra
- Comment: The value includes the network name and address

printers

In most cases, the printers map is created from the /etc/printers.conf file. A typical line looks like this:

destination_name key=value[,key=value,...] # comment

For example:

```
buildx:paddr=buildx.acme.com,105004,1,sys,lp,buildxspl,1:
printer3:bsdaddr=server,ps_printer # in copy room
```

source

```
/etc/printers.conf
printer3:bsdaddr=server,ps_printer # in copy room
```

AD entry

```
Key:     printer3
Value:   printer3:bsdaddr=server,ps_printer
Comment: in copy room
```

If you create a printers map in Active Directory, you must include the key as part of the value. For example:

- Key: printer3
- Value: printer name followed by key and value pairs for the printer properties: printer3:bsdaddr=server,ps_printer
- Comment: in copy room

This map is only applicable for Solaris.

prof_attr

In most cases, the prof_attr map is created from the /etc/security/prof_attr file. A typical line looks like this:

```
profile_name:res1:re2,description:attr
```

For example:

```
all:::Execute any command as the user:help=AllRights.html
guest:RO::Allow read-only:audit-flags=all:project=web
```

UNIX file

```
/etc/security/prof_attr
all:::Execute any cmd as user or role :help=U-Rhelp.html
```

NIS entry

```
prof_attr
Key:   <profile_name>
Value: <profile_name>:[res1]:[res2]:<descrip>:<attr>=<attr_value>
```

AD entry

```
Key:     all
Value:   all:::Execute any cmd as user or role :help=U-Rhelp.html
Comment:
```

If you create a prof_attr map in Active Directory, you must include the key as part of the value. For example:

- Key: all
- Value: profile name and properties followed by attributes defined as key and value pairs for the profile: all:::Execute any cmd as user or role:help=All.html

This map is only applicable for Solaris.

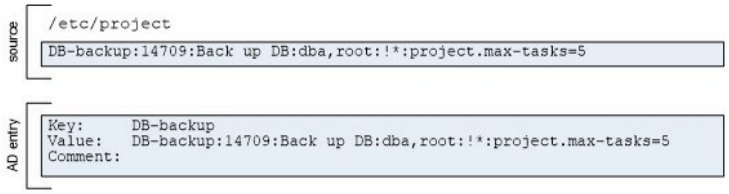
project

In most cases, the project map is created from the/etc/project file. A typical line looks like this:

```
project_name:projectid:comment:user_list:group_list:attr
```

For example:

```
DB-backup:14709:Back up DB:dba,root:!*:project.max-tasks=5
web:101:Web services deployment:root:as-team: \\ task.maxlwps=(privileged,101,signal=SIGTERM)
```



If you create a project map in Active Directory, you must include the key as part of the value. For example:

- Key: DB-backup
- Value: project name and properties followed by attributes defined as key and value pairs: DB-backup:14709:Back up DB:dba,root:!*: \\ project.maxtasks=5

This map is only applicable for Solaris.

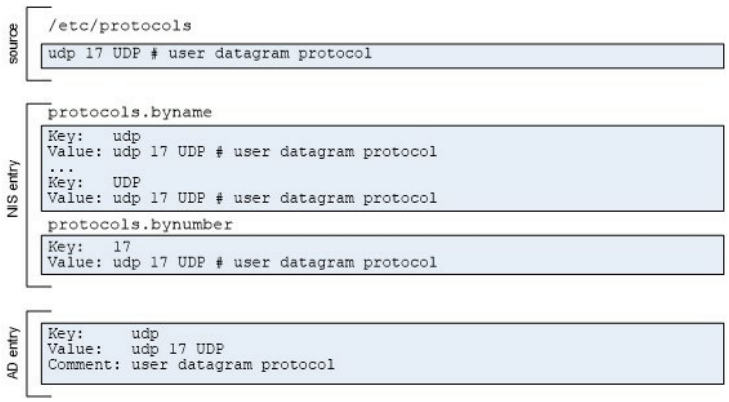
protocols

The protocols map is the the abbreviated name for the protocols.bynumber map. The derived maps are protocols.byname and protocols.bynumber. In most cases, the protocols map is created from the /etc/protocols file. A typical line looks like this:

protocol number alias # comment

For example:

ip 0 IP # internet protocol, pseudo protocol number
udp 17 UDP # user datagram protocol



For the protocols.byname map, entries are defined like this:

- Key is the protocol name: udp
- Value is the protocol name, number, and any aliases defined for the protocol: udp 17 UDP

For the protocols.bynumber map, entries are defined like this:

- Key is the protocol number: 17
- Value is the protocol name, number, and any aliases defined for the protocol: udp 17 UDP

If you create a protocols map in Active Directory, you must include the key as part of the value. For example:

- Key: udp
- Value: udp 17 UDP
- Comment: user datagram protocol

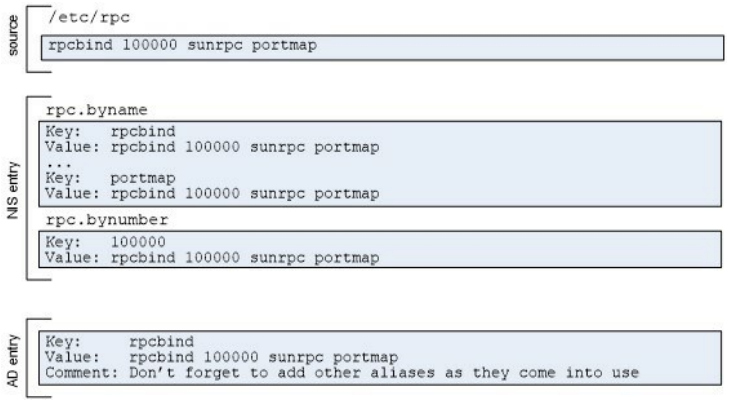
rpc

The rpc map is the the abbreviated name for the rpc.bynumber map. The derived maps are rpc.byname and rpc.bynumber. In most cases, the rpc map is created from the /etc/rpc file. A typical line looks like this:

rpc_name port_number alias1 alias2 ... # comment

For example:

```
portmapper 100000 portmap sunrpc
rpcbind 100001
```



For the rpc.byname map, entries are defined like this:

- Key is the rpc name or alias, so there would be separate entries for: portmapper, portmap, sunrpc, and rpcbind.
- Value for each of the portmapper, portmap, and sunrpc key entries would be the same: portmapper 100000 portmap sunrpc

For the protocols.bynumber map, entries are defined like this:

- Key is the rpc number: 100000
- Value is the rpc name, number, and aliases: portmapper 100000 portmap sunrpc

If you create a rpc map in Active Directory, you must include the key as part of the value. For example:

- Key: portmapper
- Value: portmapper 100000 portmap sunrpc
- Comment: portmap and sunrpc are aliases for portmapper

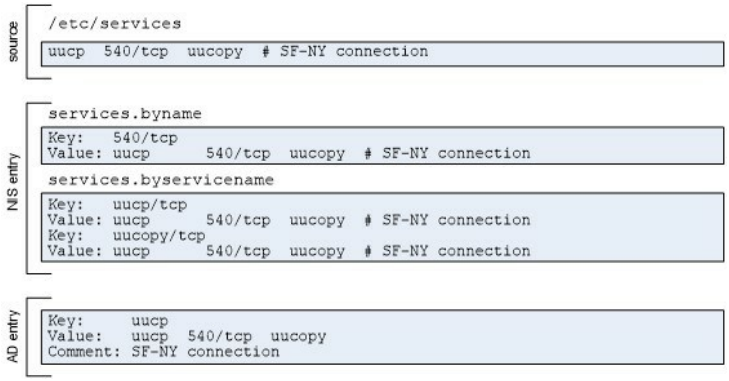
services

The services map is the the abbreviated name for the services.byname map. The derived maps are services.byname and services.byservicename. In most cases, the services map is created from the /etc/services file. A typical line looks like this:

```
service port/protocol alias1 alias2 ... # comment
```

For example:

```
uucp 540/tcp uucopy # this entry is for uucp
```



For the services.byname map, entries are defined like this:

- Key is the service name or alias, so there would be separate entries for: uucp and uucopy.
- Value for each of the uucp and sunrpc key entries would be the same: uucp 540/tcp uucopy

For the service.byservicename map, entries are defined like this:

- Key is the port number and protocol: 540/tcp
- Value contains the same set of fields: uucp 540/tcp uucopy

If you create a services map in Active Directory, you must include the key as part of the value. For example:

- Key: uucp
- Value: uucp 540/tcp uucopy
- Comment: uucopy is an alias for uucp

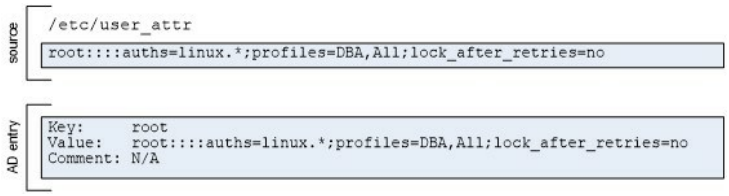
user_attr

In most cases, the user_attr map is created from the /etc/user_attr file. A typical line looks like this:

```
user:qualifier:res1:res2.attr
```

For example:

```
root::::auths=solaris.*,solaris.grant; \\
profiles=Web Console Management,All; \\ lock_after_retries=no; min_label=admin_low; \\ clearance=admin_high
```



If you create a user_attr map in Active Directory, you must include the key as part of the value. For example:

- Key: root
- Value: user name and properties followed by attributes defined as key and value pairs for the profile:

  all::::auths=solaris.*;profiles=DBA,all;lock_after_retries=no

This map is only applicable for Solaris.

Creating Generic Custom Maps

You can create generic maps to publish any type of custom information that you want to make available to NIS clients. Generic custom maps consist of a simple key/value format and optional comments. You can also use generic maps to manually create standar

To add a custom map to Active Directory:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, select **NIS Maps** and right-click; then click **New** and select **Generic Map**.
4. Type a name for the new map; then click **OK**.
5. In the details pane, select the new map, right-click; then click **New > Map entry**.
6. Type the appropriate information for the map record you are adding; then click **OK**. For example:
   - Type the **Key** to use in a client request for looking up the corresponding value.
   - Type the **Value** associated with the key.
   - Type any optional **Comments** for the key/value pair.

     For example:

Changing the Map Type

When you import or create NIS maps, the map type determines the fields defined. For example, a Generic map type consists of three fields: the **Key** field (required) the **Value** field (required), and the **Comment** field. If you don't select the correct map type, the Centrify Network Information Service will not be able to interpret the records in the map correctly or respond to client requests with the proper information.

To change the map type of an existing NIS map:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**; then the select the map name you want to change. For example, if you have created a map named nethosts, select the nethosts map.
4. Right-click; then click **Change Type** and select the correct map type. For example, if the records in nethosts map should consist of a Key, a Value, and an optional Comment, select **Generic Map** as the map type.

   If records have already been defined for the map using the incorrect map type, in most cases, you will need to modify the fields after changing the map type.

Maintaining Map Records in Active Directory

Once NIS maps are stored in Active Directory, you must maintain the records in Active Directory to ensure changes are reflected in the local map cache that the Centrify Network Information Service uses to respond to NIS client queries. You can use Access Manager to manually add, edit, or delete individual map records for any map. The specific fields available in each record, and which fields are required and which are optional, depend on the type of map you are editing. For example, the fields in an auto.master map entry are different from the fields in a netgroupmap entry. For information about the fields in different types of maps, see Creating new NIS maps in Active Directory.

Modifying Map Records in Active Directory

Specific users and groups can be given the right to add, modify, and delete NIS map entries using the Zone Delegation Wizard. For information about the rights required, see the *Planning and Deployment Guide*.

To edit individual map records:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**, the select the map you want to modify. For example, select the auto.master map.
4. Select an individual map record and right-click.
5. Click **Properties** to modify the fields for the selected record or click **Delete** to remove the record from the map.

   If deleting a map record, click **Yes** to confirm the operation.

Deleting a map stored in Active Directory

Specific users and groups can be given the right to delete NIS maps using the Zone Delegation Wizard. For information about the rights required, see the *Planning and Deployment Guide*.

To remove a NIS map from Active Directory:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**, the select the map you want to remove.
4. Right-click; then click **Delete** to remove the map from Active Directory.

This section describes how to use diagnostic tools and log files to retrieve information about adnisd operation and correct problems.

Analyzing Zones for Potential Issues

One way to avoid problems with agentless authentication or incomplete information is to periodically analyze the zone in the Active Directory forest using the **Analyze** wizard.

> **Note**: When you run the **Analyze** wizard, it checks only *open* zones in the Active Directory forest. Make sure the zone you are using as a NIS domain is open before analyzing the forest.

To check for potential problems in the Active Directory forest:

1. Open Access Manager.
2. If so prompted, specify the forest domain or domain controller to which to connect.
3. In the console tree, select the Access Manager root node, right-click, and click **Analyze**.
4. At the Welcome page, click **Next**.
5. Select the checks to perform (at least the two in the table below) and click **Next**.

Inconsistency in granting NIS server permissions	Check that a zone_nis_servers group exists in each zone that supports agentless authentication, and that the group contains all NIS servers defined for the zone (to ensure data integrity). This group is required for assigning permissions to Delinea-managed computers that act as NIS servers. Do not delete or modify it manually.
Orphan UNIX data objects	Check for profile objects whose parent objects have been deleted – for example, manually deleted zone objects whose user, group or computer UNIX profile data may be left in Active Directory. This option removes UNIX-specific data from Active Directory.

6. Review the summary report and click **Finish**.
7. If the summary report indicates any issues, select **Analysis Results** in the console tree and view the details listed in the right pane. For example:

   To drill down further, or to resolve the issue, select the warning or error, right-click, and select **Properties**. For example:

Verifying NIS Configuration for Servers and Clients

If you are troubleshooting issues with the Delinea Network Information Service or NIS client look-ups, start by verifying whether the current environment is configured properly by doing the following:

- Check the connectivity between the NIS client and the NIS server with a ping command. If the ping command fails, check the network connection and the DNS configuration for name resolution problems.
- Verify that the nisd.securenets parameter allows responses to NIS clients on other computers. By default, the adnisd process responds only to *local* NIS requests.
- Verify that the adnisd process is running, for example with the ps command. If adnisd is not running, restart it.
- Verify that ypserv is *not* currently running. If ypserv is running, stop it, modify the system initialization files so ypserv does not start when the computer is rebooted, and restart adnisd.
- Verify that adnisd has registered with RPC by running rpcinfo -p localhost on the adnisd server. You should see two entries in the RPC table for the ypserv program (100004):

100004	2	udp	844 ypserv
100004	2	tcp	846 ypserv

If no table is displayed, restart RPC services. If the ypserv process is not listed, restart adnisd.

- Verify RPC connectivity from the NIS client:

  rpcinfo -p server

  You should see the same table and entries as when you listed RPC entries for the adnisd server. For example:

100004	2	udp	844 ypserv
100004	2	tcp	846 ypserv

  If no table is displayed, check the access permissions to the RPC server. For example, on Linux, check /etc/hosts.allow and /etc/hosts.deny files.

- Make sure the correct NIS domain name is configured on the NIS client. The NIS domain name is usually the same name as the name of the zone that the server is joined to. To set the domain name, log on as root run the following command:

  domainname zone_name

- Verify that the ypbind process is running on the NIS client using the ps command. If ypbind is not listed as a running process, configure and start it.
- Verify that ypbind on the NIS client has found the Delinea NIS server by running ypwhich on the NIS client machine.

  If the client is not bound to the correct server name, check the ypbind configuration files and start-up options.

  If you are transitioning from an existing NIS infrastructure to the Delinea Network Information Service, the most common reasons for errors are an incorrect domainname setting or an improper ypbind configuration. For example, if your existing NIS domain names do not match the zone name, some clients may fail because they use the old NIS domain name instead of the domain name you have set up for the Delinea Network Information Service domain.

Updating the Startup Sequence

On some platforms, the adnisd package might prevent the ypbind service from starting properly because of the order in which services are started. For example, if ypbind is configured to start before the adnisd service, the bind will fail. In most cases, this issue does not occur if you are installing new packages because the installation process checks and corrects the startup sequence to ensure that the bind will be successful. However, to prevent unintended changes to the existing startup sequence during an upgrade, upgrading the adnisd package will not modify your existing startup configuration. You can manually correct the startup sequence after an upgrade by running the chkconfig script. For example, run the following command after the adnisd upgrade:

chkconfig adnisd on

Using NIS Command Line Utilities

The Delinea Network Information Service supports common command-line utilities for performing administrative and diagnostic tasks. The following table lists those you may find useful in the Delinea NIS environment.

ypwhich	Display the name of the NIS server the client is connected to.
ypwhich -m	List the maps that are served by the current NIS server.
ypwhich -x	Display the nicknames that are defined for NIS maps.

ypcat -k map	Display the contents of the specified map. This command displays both keys and values.
ypmatch key map	Look-up the specified key in the specified map.
yppoll map	Check the version number of the specified map. This command is only available on Solaris and HP-UX environments. The version number is displayed as an integer. The adnisd process does not use timestamps.

Configuring Logging for adnisd

By default, the adnisd process logs errors, warnings, and informational messages in the syslog and /var/log/messages files, along with other kernel and program messages. You might find it useful to log additional details about the operation of the adnisd process for troubleshooting purposes.

To enable logging for the Delinea Network Information Service:

1. As root, set the logging level for the Delinea Network Information Service by modifying the log.adnisd parameter in the centrifydc.conf file.

   You might also want to suppress log messages from adclient to make it easier to collect and analyze the messages that are specific to adnisd operation. For example, set the log.adnisd parameter to DEBUG to log *all* adnisd operations, and the log parameter for adclient to INFO or WARN to limit messages generated by the adclient process:

   ```
   log: WARN log.adnisd: DEBUG
   ```

   If you only want to collect diagnostic information for netgroup processing, set the log.adnisd.netgroup parameter instead of the log.adnisd parameter. For example:

   ```
   log.adnisd.netgroup: DEBUG
   ```

2. Set the syslog facility to use for logging adnisd operations using the logger.facility.adnisd configuration parameter. This parameter enables you to log adnisd messages using a different syslog facility than the facilities used for logging general adclient messages or adclient audit messages.

   This parameter value can be any valid syslog facility. For example, set this parameter to log messages to auth (default), authpriv, daemon, security, or local0-7 facilities. For example:

   ```
   logger.facility.adnisd: auth
   ```

For performance and security reasons, only enable DEBUG logging when necessary – for example, when requested to do so by Delinea Support, or while diagnosing a problem.

> **Note**: Sensitive information may be written to this file. Evaluate the contents before giving others access to it.

Smart Card for Red Hat Linux

This document explains how to set up smart card authentication for logging on to Red Hat Linux computers.

Why and How to Use a Smart Card to Log On

Smart cards provide an enhanced level of security for Red Hat Linux computers when users log on to Active Directory domains. If you use a smart card to log on, authentication requires a valid and trusted root certificate or intermediate root certificate that can be validated by a known and trusted certification authority (CA).

Because smart cards rely on a public-private key infrastructure (PKI) to sign and encrypt certificates and validate that the certificates were issued by a trusted certification authority and have not expired or been revoked, authentication using a smart card is more secure than a user name and password.

Configuring a smart card for use on a Red Hat Linux computer that is running the Delinea Agent requires that you have already set up a smart card for use in a Windows domain. You do not need to add any smart card infrastructure to the Linux computer, other than a smart card reader and a provisioned smart card.

In a Windows environment, a smart card may be set up either for a single user account or for multiple user accounts. For example, an individual contributor might have access to a single Active Directory account that he uses for all his work. In this case, the card is set up for a single user and the card is linked directly to a UPN. When a user inserts the card to log on, the smart card system looks for the UPN in Active Directory and prompts for a PIN.

Windows 2008 also provides a name-mapping feature that enables configuring a smart card with multiple user accounts. For example, a user might want to log in with a regular account to check mail or perform routine tasks, but log in with an administrator's account to perform privileged tasks. To set up a card for multiple users, an administrator maps a certificate to each user account on the card. When a user inserts the card to log on, the smart card system prompts the user to select which account to use, and prompts for the card's PIN.

If you have set up smart card login for Windows clients in a domain, you can use Access Manager to configure smart card login for Red Hat Linux clients joined to the same domain. If you have provisioned a smart card for use on a Windows computer — either for a single user or multiple users — once you configure smart card support for a Linux computer, you can use the same smart card to log in to a Red Hat Linux computer.

> **Note**: Configuring smart card support in Access Manager is nearly the same for a single-user or multi-user card with the exception that for multi-user cards, you must set an extra configuration parameter as explained in Enabling Support for Multi-User Smart Cards.

Setting up a single user smart card login for Windows computers requires either:

- Microsoft enterprise root certification authority; see the Microsoft TechNet article: Install an enterprise root certification authority.
- A third party certification authority — see the Microsoft KB article: Guidelines for enabling smart card logon with third-party certification authorities.

Setting up a multi-user smart card login for windows requires mapping the certificate on the card to the users who the card is associated with. See the following Microsoft Technet Blog post: Mapping One Smart Card to Multiple Accounts for more information on how to do this.

Configuring Smart Card Authentication

You configure Red Hat Linux computers for smart card authentication primarily through group policy settings. Enabling support for smart cards requires that you set a single policy ("Enable smart card support"). Supporting the use of multi-user smart cards requires that you set a configuration parameter on each Red Hat computer. In addition, Server Suite provides several group policies to control how smart card authentication works after you enable it.

Complete the procedures in the following sections to configure smart card authentication for Red Hat Linux computers:

- Enabling smart card support in which you enable smart card authentication for Active Directory users. This is the only procedure you need to complete to enable smart card authentication. The other procedures allow you to configure different aspects of smart card authentication, such as locking the screen if the smart card is removed, or preventing users from logging in without a smart card.
- Enabling support for multi-user smart cards in which you set the smart card.name.mapping configuration parameter to enable the use of smart cards provisioned with multiple users on a particular computer.
- Enforcing smart card authentication in which you prevent users from logging in with a user name and password on Red Hat Linux computers that have smart card authentication enabled. You can require all users on a computer to use a smart card for logging in or require specific users to use a smart card.
- Configuring certificate validation in which you specify how to use a Certificate Revocation List (CRL) to check the status of certificates stored on a revocation server
- Locking the screen if a smart card is removed in which you require that the computer's screen is locked when a smart card is removed.
- Enabling a certificate without extended key usage in which you enable a Windows group policy setting to allow using certificates without the EKU attribute for smart-card log in.
- Configuring applications for smart card access in which you configure applications such as Firefox and Thunderbird that require smart card authentication to gain access to sensitive sites and data.

Before Configuring Smart Card Authentication

To use a smart card to log on to a Red Hat Linux, CentOS, Debian, or Ubuntu computer, verify that the computers meet these requirements:

- Are running one of the following operating systems:
  - Red Hat Linux (32- or 64-bit) version 5.6 or later
  - CentOS version 5.6 or later
  - Ubuntu 18.04.x LTS, 20.04.x LTS and 21.04 (amd64)
  - Debian 9.x and 10.x (amd64)
  - Rocky Linux (amd64)
  - AlmaLinux (amd64)

    Note: For Debian and Ubuntu systems, be sure to have the opensc-pkcs11, pcscd, and libnss3-tools packages installed.
- Are running the GNOME desktop. The agent does not support use of a smart card with the KDE desktop.
- If a system is running RedHat Linux or CentOS 8.0 or later, the system needs Delinea Agent for *NIX version 5.7.0 or later.
- If a system is running Debian or Ubuntu, the system needs Delinea Agent for *NIX version 5.8.0 or later.
- Are joined to the Windows domain.
- Have a supported smart card reader attached.

Other prerequisites for enabling smart card support differ depending on whether you have configured a single-user or multi-user smart card.

For a single-user card, before enabling smart card support, make sure you do the following:

- Provision a smart card with an NT principal name and PIN. Currently, Access Manager supports Common Access Card (CAC), Personal Identify Verification (PIV), cards with both CAC and PIV profiles (CACNG), and Alternative Logon Token (ALT) smart cards.
- Verify that the Active Directory Zone user's UPN matches the UPN on the smart card.

For a multi-user card, before enabling smart card support, make sure you have the following in place:

- A Windows Server 2008, or later, domain controller for authentication.
- The card is not configured with a UPN. If a card with a UPN is inserted, the computer prompts for a PIN rather than prompting for a user name and password.
- An administrator has added the certificate on the card to the name mapping for the users the card is associated to. See the following Microsoft Technet Blog post: Mapping One Smart Card to Multiple Accounts for more information on how to do this.

For either type of card, verify that the public key infrastructure to support smart card login is operational on the Windows computer running Active Directory and Access Manager. If the user is able to log in to a Windows computer with a smart card, and you have a card reader and a fully-provisioned card for the Linux computer, the user should be able to log in to the Linux computer once you configure it for smart card support.

Although the Linux computer has its own infrastructure for enabling and managing smart card authentication, the Delinea Agent for *NIX and smart card utility (sctool) enable authentication through Active Directory. After you enable smart card support through the Delinea Agent, the Red Hat smart card configuration options have no effect.

Enabling Smart Card Support

Smart card authentication requires configuration changes to certain Red Hat or CentOS Linux files, depending on the version of Red Hat Linux or CentOS you are using.

For example, if you are using Red Hat Linux 5.6 or 6.0, the files affected may include the following:

- /etc/pam.d/gdm
- /etc/pam.d/gnome-screensaver
- /etc/pam.d/password-auth
- /etc/pam.d/smartcard-auth

Smart card authentication also requires configuration changes to certain system Coolkey symbolic links such as the following:

- /usr/lib(64)/libckyapplet.so.1.0.0
- /usr/lib(64)/pkcs11/libcoolkeypk11.so

After you enable smart card authentication, the agent makes the required changes and creates backup copies of the affected files.

The smart card components on the Linux computer are configured by default to use the Delinea Coolkey PKCS #11 module for authentication. Although this is the optimal configuration, if your smart cards are not supported by Coolkey, Delinea allows you to specify a different PKCS #11 module to use for authentication. Delinea does not supply PKCS #11 modules other than the default Coolkey module. If you need to use a third-party module, you must install it yourself.

Some PKCS #11 modules may not work seamlessly with the GDM environment. For example, some card events, such as locking the screen upon card removal, may not work.

To configure a different module, do one of the following:

- If you are enabling smart card support with group policy, you can specify an alternate PKCS #11 module when you enable the group policy; see the procedure: To enable smart card support by using group policy.
- If you are manually enabling smart card support by running sctool, you can set a configuration parameter on each Linux computer to specify the module to use; see the procedure: To manually enable smart card and specify a different PKCS .

Steps

If you are running Red Hat Linux 6.0, you must install some support packages before enabling smart card support; see To install required packages on Red Hat Linux 6.0.

You can enable smart card authentication by either of the following methods:

- Use the "Enable smart card support" group policy, which enables smart card support on all computers to which the Group Policy object applies. Note that configuration changes do not take place until the next group policy update or when you run adgpupdate on the Linux computers.
- Run the sctool -enable utility on each computer that you want to enable for smart card support.

To install required packages on Red Hat Linux 6.0

1. Log on to a Red Hat computer with root privilege and open a terminal window.
2. Run the following command:

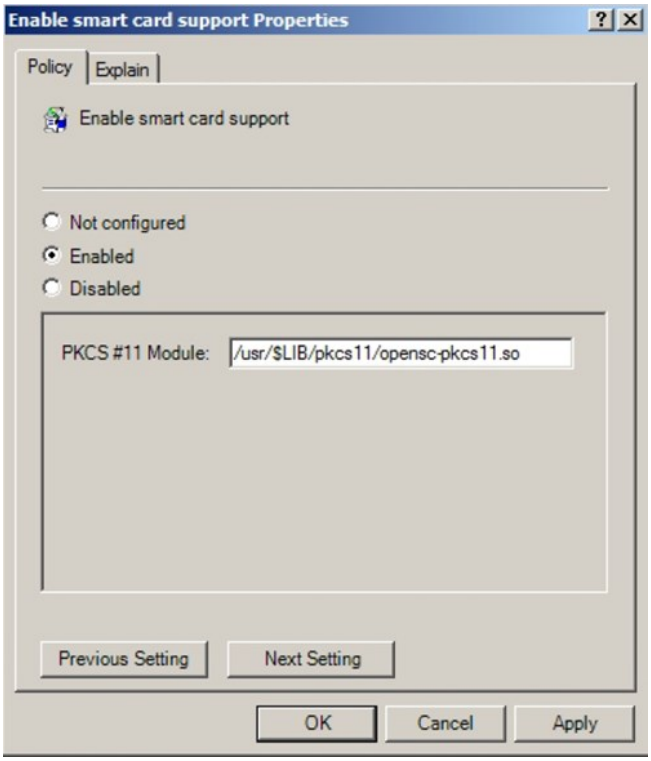
```
[root]#yum groupinstall "Smart card support"
```

To Enable Smart Card Support Using Group Policy

1. On a Windows computer, open Group Policy Management to create or select a Group Policy object that is linked to a site, domain, or organizational unit that includes Red Hat Linux computers; right-click the Group Policy object, then select **Edit**.
2. In the Group Policy Management Editor, expand **Computer Configuration** › **Policies** › **Delinea Settings** › **Linux Settings**, click **Security**, then double-click **Enable smart card support**.
3. Select **Enabled**, then click **OK** to save the policy setting, or go to the next step to change the PKCS #11 module used for authentication.

This group policy modifies Red Hat Enterprise Linux configuration files to look for a smart card user's credentials in Active Directory and verify the identity of the user with the smart card certificate.

4. Optionally, to specify a PKCS #11 module other than the Delinea default module, type the complete path to the module in **PKCS #11 Module**:



> **Note**: Your smart card environment performs optimally when configured to use the default Coolkey module. You should specify a different module only if your smart cards are not supported by Coolkey. Otherwise, skip this step and click **OK** to save the group policy setting.

This field supports the use of the $LIB environment variable in the path to allow a single group policy to work for 32-bit and 64-bit systems. At run time on 32-bit systems$LIBresolves tolib, while on 64-bit systems it resolves to lib64.

For example, the following path specifies the OpenSC PKCS #11 module:

```
/usr/$LIB/pkcs11/opensc-pkcs11.so
```

5. To apply the group policy immediately to any computer you must restart the computer or run the adgpupdate command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval. After computers are restarted or receive the policy update, they are ready for smart card use.

To Manually Enable Smart Card Support Running sctool

1. Log on to a Red Hat computer with root privilege and open a terminal window.
2. Run the sctool utility with the --enable option:

   ```
   [root]$ sctool --enable
   ```

3. Repeat steps 1 and 2 for each computer on which to enable smart card authentication.

To Manually Enable Smart Card and Specify a Different PKCS

1. Open the Delinea configuration file with a text editor, find the rhel.smartcard.pkcs11.module parameter, and set its value to the complete path for your PKCS #11 module.

Be certain to remove the comment for the parameter.

For example, the following parameter value sets PKCS #11 to the OpenSC module:

```
[user]$ vi /etc/centrifydc/centrifydc.conf
...
rhel.smartcard.pkcs11.module: /usr/$LIB/pkcs11/opensc-pkcs11.so
```

This parameter supports the use of the $LIB environment variable in the path to allow a single path specification to work for 32-bit and 64-bit systems. At run time on 32-bit systems $LIB resolves to lib, while on 64-bit systems it resolves to lib64.

2. Save and close the file.
3. Enable, or re-enable smart card support by running the following sctool commands as root:

```
[root]$ sctool --disable
[root]$ sctool --enable
```

4. Refresh the GNOME environment by running the following command as root:

```
[root]$ /usr/sbin/gdm-safe-restart
```

Next Steps

After you enable smart card support, the computer is ready for smart card authentication. You can attach a smart card reader and log in with a valid card and matching Active Directory user.

The next step is to configure one or more of the following smart card authentication options if you wish:

- Enabling support for multi-user smart card which sets the smartcard.name.mapping configuration parameter to enable the use of smart cards provisioned with multiple users on a particular computer.
- Enforcing smart card authentication which prevents users from logging on with just a user name and password.
- Configuring certificate validation which specifies how certificates are validated.
- Locking the screen if a smart card is removed which locks the screen when a smart card is removed to provide enhanced security.

If you have no other options to configure, you can go directly to Verifying smart card authentication to confirm that you can log on to one of the Linux computers that you have configured for smart card authentication.

Enabling Support for Multi-User Smart Cards

If you plan to use multi-user smart cards on a Red Hat Linux computer in your domain, you must set the smartcard.name.mapping parameter to true in the Delinea configuration file for that computer by completing the following the procedure. If your environment exclusively uses single-user smart cards, you can skip this section.

> **Note**: Setting the configuration parameter with this procedure has no effect on single-user smart cards. There is no conflict with using single-user and multi-user on the same computer. However, if a Red Hat Linux computer is accessed through a multi-user card, you must set the configuration parameter by using this procedure.

To enable support for multi-user smart cards

1. On the Red Hat Linux computer, open the Delinea configuration file in a text editor, /etc/centrifydc/centrifydc.conf, with a text editor.
2. Type the following:

   ```
   smartcard.name.mapping: true
   ```

   By default, this parameter is set to false and the configuration file should have a commented line showing this setting. So, alternately, you can find this parameter in the file, remove the comment, and change the value to true.
3. Save and close the file.

Enforcing Smart Card Authentication

By default, enabling smart card support does not force all users to log on using a smart card. If you want to require all Active Directory users to authenticate by using a smart card, you have the option to configure a computer group policy. If you want to require only specific Active Directory users to authenticate by using a smart card, you can configure their user account properties to require a smart card for authentication.

You can enable the "Require smart card login" group policy to ensure that all Active Directory users logging on to a computer must insert a smart card for authentication. If you enable this policy, Active Directory users who forget their smart card will be unable to log on to their computers. However, you add exceptions to this group policy to allow users who forget their smart card to log on using their user name and password on the computers where the policy with exceptions is applied.

> **Note**: If you use this approach to enforce smart card login for all users, be certain that all users have their accounts set with the "Password never expires" option. If a user attempts to log on with a smart card but the password for the account has expired, the smart card login fails with an error message about changing the password. If you use the account option to require smart card for specific users, you can ignore password expiration.

Enforcing smart card authentication applies to all forms of log on, including GUI login, SSH, telnet, and so on. However, it is enforced for Active Directory users only. If a computer is configured with one or more local accounts, those accounts are still able to log on even if you set the group policy to require smart card authentication.

Steps

To require smart card login, complete one of these procedures

- To require smart card login for all users on a computer
- To require smart card login for a specific user

To require smart card login for all users on a computer

1. On a Windows computer, open Group Policy Management and select the Group Policy object where you enabled smart card support for Red Hat Linux computers; right-click the Group Policy object, then click **Edit**.
2. In the Group Policy Management Editor, expand **Computer Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Require smart card login**.
3. Select **Enabled**.

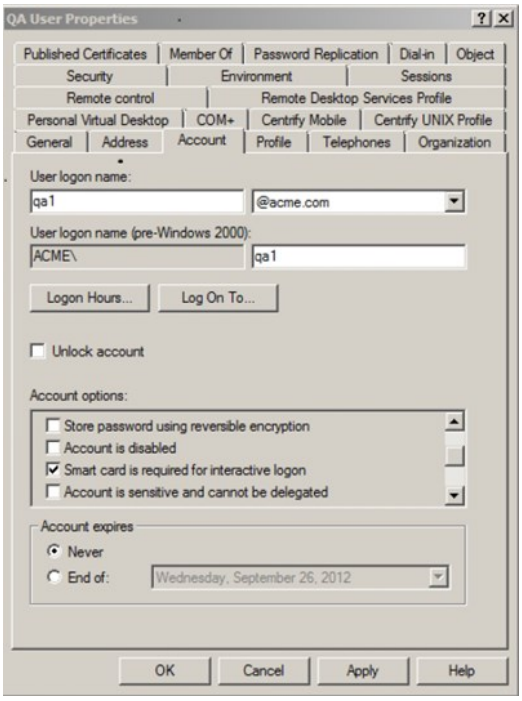
   Click **Add** if you want to add exceptions to this group policy now, then click Browse to search for and select the Active Directory group allowed to log on using a user name and password if they forget their smart card. If you only want to configure exceptions when they are needed, click **OK** to enable the group policy without exceptions.
4. To apply the group policy immediately to any computer, you must restart the computer or run the adgpupdate command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval.

To require smart card login for a specific user

1. On a Windows computer, open the Access Manager console or Active Directory Users and Computers.
2. Select the user.

   For example, in the Administrator's Console, open *domainName* ___› **Zones** ›** *zoneName* **› **UNIX Data** › **Users**.
3. Right-click the user's name and select **AD Properties**.
4. In the User Properties window for the user, click the **Account** tab.
5. In "Account options", scroll until **Smart card is required for interactive logon** is visible, then select it.


QA User Properties
Published Certificates
Member Of
Password Replication
Dial-in
Object
Security
Environment
Sessions
Remote control
Remote Desktop Services Profile
Personal Virtual Desktop
COM+
Centrify Mobile
Centrify UNIX Profile
General
Address
Account
Profile
Telephones
Organization
User logon name:
qa1
@acme.com
User logon name (pre-Windows 2000):
ACME\
qa1
Logon Hours...
Log On To...
Unlock account
Account options:
Store password using reversible encryption
Account is disabled
Smart card is required for interactive logon
Account is sensitive and cannot be delegated
Account expires
Never
End of:
Wednesday, September 26, 2012
OK
Cancel
Apply
Help

6. Click **OK**.

Configuring Certificate Validation

You can use the **Certificate validation method** group policy to configure how certificates are validated or rejected by using a Certificate Revocation List (CRL) stored on a revocation server.

To Configure How Certificates are Validated

1. On a Windows computer, open Group Policy Management and select the Group Policy object where you enabled smart card support for Red Hat Linux computers; right-click the Group Policy object, then click **Edit**.
2. In the Group Policy Management Editor, expand **Computer Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Certificate validation method**.
3. Select **Enabled**.
4. Choose one of the following options from **Certificate Revocation List**:
    - **Off:** To disable certificate validation.

      If you select this setting, no revocation checking is performed.
    - **Best attempt:** To check that certificates are not rejected as invalid, untrusted, or revoked by the certificate revocation list (CRL).

      This setting is appropriate for most organizations.
    - **Require if cert indicates:** To check whether there is a successful connection to the revocation server.

      If a URL to the revocation server is provided in the certificate, this setting requires a successful connection to a revocation server, and checks that certificates are not rejected as invalid, untrusted, or revoked by the CRL. You should only use this setting in a tightly controlled environment that guarantees the presence of a CRL server. If a CRL server is not available, certificate validation may prevent furthering processing of an authentication request.
    - **Require for all certs:** To require successful validation of all certificates.

      You should only use this setting in a tightly controlled environment that guarantees the presence of a CRL server. If a CRL server is not available, certificate validation may prevent furthering processing of an authentication request.
5. Click **OK** to save the policy settings.
6. To apply the group policy immediately to any computer, restart the computer or run the `adgpupdate` command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval.

Locking Screen if Smart Card is Removed

Depending on what you consider best practices for using a smart card, you may want the screen to lock whenever a user removes the smart card. If you want to lock the screen when a smart card is removed, you can do so by enabling the **Removing a smart card locks screen** user group policy.

To lock the smart card screen when a smart card is removed

1. On a Windows computer, open Group Policy Management and select the Group Policy object where you enabled smart card support for Red Hat Linux computers; right-click the Group Policy object, then click **Edit**.
2. In the Group Policy Management Editor, expand **Computer Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Lock Smart Card screen for RHEL**.
3. Select **Enabled**, then click **OK**.

   Note: Policies are turned off by default on Linux systems but can be turned on with a group policy setting. To ensure that the "Removing a smart card locks screen" policy takes effect, verify that the following computer policy is enabled by completing the following two steps.
4. Expand **Computer Configuration** › **Centrify Settings** › **DirectControl Settings**, click **Group Policy Settings**, then double-click **Enable user group policy**.
5. Verify that **Enabled** is selected, and if not, select it, then click **OK**.
6. To apply the group policy "Lock Smart Card screen for RHEL" immediately to any computer you must restart the computer or run the `adgpupdate` command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval. After computers are restarted or receive the policy update, the screen is locked if a smart card is removed.

Enabling a Certificate Without Extended Key Usage

Normally, smart card use requires certificates that contain the extended key usage (EKU) attribute. However, Windows provides a group policy that allows the use of certificates that do not have the EKU attribute.

> **Note**: This group policy is implemented as an administrative template (.adm file), not as an xml file, as are the Delinea group policies.

To use certificates without the EKU attribute with smart cards:

1. Open the group policy editor and edit the GPO that contains the Linux computers enabled for smart-card login.
2. Open **Computer Configuration > Policies > Administrative Templates > Windows Components > Smart Card** and double-click **Allow certificates with no extended key usage certificate attribute**.
3. Click **Enabled** and click **OK**.

   When you enable this policy, it sets the smartcard.allow.noeku parameter to true in the Delinea configuration file. Certificates with the following attributes can also be used to log on with a smart card:

   - Certificates with no EKU
   - Certificates with an All Purpose EKU
   - Certificates with a Client Authentication EKU

4. In a Terminal window, run the sctool command as root with the -E (--no-eku) parameter to re-enable smart card support. You must use either the -a (--altpkinit) or -k (--pkinit) parameter with the -E option; for example:

   ```
   sctool -E -k jsmart@acme.com
   ```

Configuring Applications for Smart Card Access

Many applications, including Firefox and Thunderbird, that require smart card access to sensitive sites or data, create their own NSS database for the user. To give these applications access to the certificates and control revocation lists (CRL) used by the agent for log on, you enable the group policy "Specify applications to import system NSSDB", which synchronizes the system NSSDB file on a computer with each application's NSSDB file.

Each application, such as Firefox, creates a profile file (profile.ini) that specifies the location for its certificates and CRLs. With the "Specify applications to import system NSSDB" policy, you specify the location of the profile file for an application. A Delinea mapper file parses the profile file to determine the location of the application's certificates and CRLs and copies certificates and CRLs to this location.

Steps

If the computers you manage use applications such as Firefox that require smart card access to sensitive sites or data, configure NSS database synchronization to ensure that these applications have access to current certificates and control revocation lists.

To configure NSS database synchronization

1. On a Windows computer, open Group Policy Management and select the Group Policy object where you enabled smart card support for Red Hat Linux computers; right-click the Group Policy object, then select **Edit**.
2. In the Group Policy Management Editor, expand **User Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Specify applications to import system NSSDB**.
3. Select **Enabled**, then click **Add**.
4. In **Application**, specify the application directory in which to import the system NSS database.

   For each application enter the location of its profiles.ini file. Specify the entry in relation to the home directory of the user by starting the path with ~/. For example, the following entry specifies the default location of the Firefox profiles.ini file

   ~/.mozilla/firefox.
5. Click **Add** to add as many application directories as necessary, then click **OK** to save the settings.

   > **Note**: User policies are turned off by default on Linux systems but can be turned on with a group policy setting. To ensure that the "Specify applications to import system NSSDB" policy takes effect, verify that the following computer policy is enabled:
6. Expand **Computer Configuration** › **Centrify Settings** › **DirectControl Settings**, click **Group Policy Settings**, then double-click **Enable user group policy**.
7. Verify that **Enabled** is selected, and if not, select it, then click **OK**.
8. To apply the group policy immediately to any computer, restart the computer or run the `adgpupdate` command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval. After computers are restarted or receive the policy update, the screen is locked if a smart card is removed.

Configuring Citrix VDA Smart Card Authentication

You can integrate Delinea Agent for *NIX with the Citrix Virtual Delivery Agent (VDA) for Active Directory user authentication. This integration helps users log in to remote Red Hat Linux (RHEL) virtual desktop sessions with a smart card connected to the client device.

The Delinea Authentication Service supports pass-through authentication if the Citrix requirements are met. For details, see https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/system-requirements.html.

Be sure that you have set up smart card authentication already on Windows systems in your domain before continuing.

To configure Citrix VDA smart card authentication:

1. Install the Citrix Linux VDA.

   For details, see the Citrix documentation: https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/installation-overview.html.

   For example, you might run a command that looks like the following:

   ```
   sudo yum -y localinstall XenDesktopVDA-19.9.0.3-1.el7_x.x86_64.rpm
   ```

2. According to the Citrix VDA documentation, install the necessary software and perform the required system integrations.

   NTP isn't required to be configured in an Active Directory environment, but Citrix Linux VDA does require certain software, such as PostgreSQL and openJDK. For details, see https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/configuration.html.

3. Install Delinea Agent for *NIX version 19.9 or later and join the computer to the domain.

   For details, see the **Planning and Deployment Guide**.

4. To configure the agent integrations with the Citrix Linux VDA, add the following setting to the centrifydc.conf file:

   ```
   smartcard.login.service.accounts: ctxsrvr
   ```

   The Citrix smart card login service runs as the ctxsrvr account. This parameter allows you to specify a list of non-root user accounts that use the smart card login services.

5. Configure the Citrix Linux Virtual Delivery Agent (VDA).

   For details, see the Citrix documentation: https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/configuration.html and https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/installation-overview/redhat.html.

   Below is an example of running ctxsetup.sh in interactive mode; be sure to adjust as needed for your environment.

   ```
   $ sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
    Welcome to the Citrix Linux VDA setup script. This script will guide you through the
    configuration of the Linux VDA system services. You can re-run this script at
    any time to reconfigure the system.
    Gathering information...
    Checking CTX_XDL_DOTNET_RUNTIME_PATH... Value not set.
    Dotnet Core runtime environment is needed to run Linux VDA.
    Linux VDA will install it to /opt/dotnet by default.
    If required, please specify an absolute path in valid format here (e.g., /the/path). [\<none\>]:
    Checking CTX_XDL_SUPPORT_DDC_AS_CNAME... Value not set.
    The Virtual Delivery Agent supports specifying a Delivery Controller name using a DNS CNAME record.
    Do you want to enable support for DNS CNAME records? (y/n) [n]: y
    Checking CTX_XDL_DDC_LIST... Value not set.
    The Virtual Delivery Agent requires a space-separated list of Delivery Controller Fully Qualified Domain Names
    (FQDNs) to use for registering with a Delivery Controller. Please provide the FQDN of at least one Delivery
    Controller: CS.CITRIX.TEST
    Checking CTX_XDL_VDA_PORT... Value not set.
    The Virtual Delivery Agent by default communicates with Delivery Controllers using TCP/IP port 80.
    Please provide the TCP/IP port the Virtual Delivery Agent service (ctxvda) should use to communicate with a
    Delivery Controller [80]:
    Checking CTX_XDL_REGISTER_SERVICE... Value not set.
    The Linux VDA services support starting during boot.
    Do you want to register these services to start on boot? (y/n) [y]:
    Checking CTX_XDL_ADD_FIREWALL_RULES... Value not set.
    The Linux VDA services require incoming network connections to be allowed through
    the system firewall. Do you want to automatically open the required ports (by default ports 80, 1494, 2598, 8008 and 6001\~6099) in the
    system firewall for the Linux VDA? (y/n) [y]:
    Checking CTX_XDL_AD_INTEGRATION... Value not set.
    The Virtual Delivery Agent requires Kerberos configuration settings to authenticate with Delivery Controllers. The
    Kerberos configuration is determined from the installed and configured Active Directory integration tool on this
   ```

```
system. Please select the Active Directory integration tool configured on this system:
1: Winbind
2: Quest
3: Centrify 4: SSSD
5: PBIS
Select one of the above options (1-5) [1]: 3
Checking CTX_XDL_HDX_3D_PRO... Value not set.
Linux VDA supports HDX 3D Pro, a set of graphics acceleration technologies designed to optimize the
virtualization of rich graphics applications. HDX 3D Pro requires a compatible NVIDIA Grid graphics card to be
installed. If HDX 3D Pro is selected the Virtual Delivery Agent will be configured for VDI desktops (single-session)
mode. Do you want to enable HDX 3D Pro? (y/n) [n]:
Checking CTX_XDL_VDI_MODE... Value not set.
Linux VDA supports delivery of hosted shared desktops (multi-session) or VDI desktops (single-session).
Do you want to enable VDI desktops (single session) mode? (y/n) [n]: y
Checking CTX_XDL_SITE_NAME... Value not set.
The Virtual Delivery Agent discovers LDAP servers using DNS, querying for LDAP service records. To limit the DNS
search results to a local site, a DNS site name may be specified.
If required, please specify a local DNS site name. [\<none\>]:
Checking CTX_XDL_LDAP_LIST... Value not set.
The Virtual Delivery Agent by default queries DNS to discover LDAP servers, however if DNS is unable to provide
LDAP service records, you may provide a space-separated list of LDAP Fully Qualified Domain Names (FQDNs) with
LDAP port (e.g. ad1.mycompany.com:389).
If required, please provide the FQDN:port of at least one LDAP server. [\<none\>]:
Checking CTX_XDL_SEARCH_BASE... Value not set.
The Virtual Delivery Agent by default queries LDAP using a search base set to the root of the Active Directory
Domain (e.g. DC=mycompany,DC=com), however to improve search performance, a search base may be specified
(e.g. OU=VDI,DC=mycompany,DC=com).
If required, please provide an LDAP search base. [\<none\>]:
Checking CTX_XDL_FAS_LIST... Value not set.
The Federated Authentication Service (FAS) servers are configured through AD Group Policy. But because
the Linux VDA does not support AD Group Policy, you can provide a semicolon-separated list of FAS servers instead.

Caution 1: The sequence must be the same as configured in AD Group Policy.

Caution 2: If any server address is removed, you must fill its blank with the '\<none\>' string and keep the index of server addresses without any changes.
If required, please specify the list of FAS servers (e.g., fasserver.company.com). [\<none\>]:
Checking CTX_XDL_START_SERVICE... Value not set.
The Linux VDA services may be started after configuration is complete.
Do you want to start these services once configuration is complete? (y/n) [y]:
Configuring Citrix Linux VDA ...
Configuration complete.
```

6. In Citrix Virtual Apps or Citrix Virtual Desktops, create the machine catalog and delivery group.

   For details, see https://docs.citrix.com/en-us/linux-virtual-delivery-agent/current-release/installation-overview/redhat.html#step-8-create-the-machine-catalog-in-citrix-virtual-apps-or-citrix-virtual-desktops.

7. (Optional) Enable the group policy entitled "Enable smart card support."

8. Verify that the smart card login is enabled on the Linux computer:

   ```
   $ sudo sctool -s
   ```

   Delinea Smart Card support is enabled.

   If you have not enabled the group policy enabled "Enable smart card support", you may need to run the following command to enable smart card login:

   ```
   $ sctool -e
   ```

   For details about this group policy, see the **Smart Card Configuration Guide**.

9. Reboot the Linux computer.

10. In Citrix StoreFront, enable smart card authentication.

    For details, see https://docs.citrix.com/en-us/storefront/current-release/configure-authentication-and-delegation/configure-authentication-service.html.

Verifying Smart Card Authentication

After you enable smart card support, you should verify that a user is able to authenticate with a smart card on a Red Hat Linux computer.

To verify smart card authentication:

1. On the Red Hat Linux computer, run the following command to check the status of smart card support:

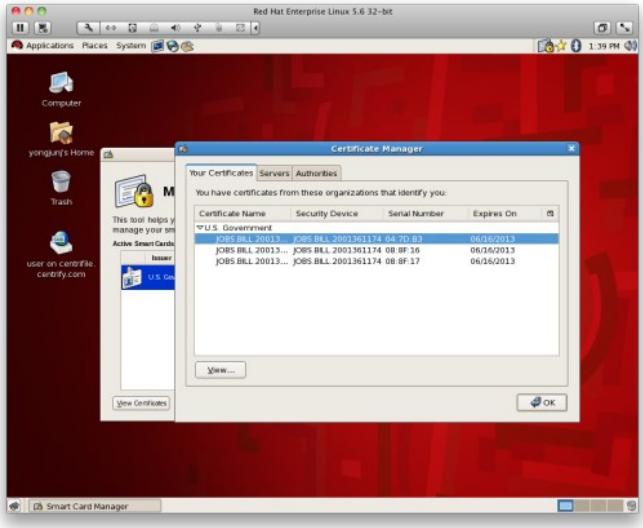
   [root]#sctool --status DirectControl Smart Card support is enabled.

   > **Note**: On Red Hat Linux computers, when enabling smart card support, the agent bypasses the native, Red Hat, smart card infrastructure. Therefore, after you enable smart card with the agent (through the group policy setting or the sctool command), the sctool --status command will show that smart card is enabled but the Red Hat system (GNOME: System > Administration > Authentication > Authentication) might show that it is not enabled. You can ignore the GNOME setting because it is for native smart card authentication, not the authentication used by the agent.

2. Click **System > Administration > Smart Card Manager**.

3. Insert the smart card in the reader and click **View Certificates**.

4. Double-click the certificate for a user account that has a profile in the zone the Red Hat Linux computer has joined, for example, **JOBS.BILL.20013**.



5. Scroll to find the NT Principal name; for example:

NT Principal Name jbill.20013@myDomain.com

6. On a Windows computer, open Activity Directory Users and Computers or the Access Manager console. For example, in the Access Manager console, navigate to the zone that the Red Hat Linux computer has joined and open **UNIX Data > Users**, then double-click the user.

   The NT Principal name in the certificate should match the login name in the Delinea UNIX profile, or in the Active Directory Account tab.

7. Log out of the Red Hat computer.

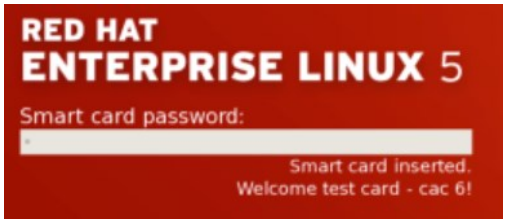
8. Re-insert the smart card in the reader and enter the user's PIN.

Using a Smart Card at Login

When a user inserts a smart card into the card reader attached to a Red Hat Linux computer that is waiting for login, the login dialog is replaced by a smart-card enabled login (if the card is provisioned for an Active Directory user who is enabled for the Delinea zone to which the computer is joined). However, the actual log on screen varies depending on whether the card is provisioned for a single user or for multiple users.

How the Login Screen Appears for a Single-User Card

When a user inserts a single-user card, the smart card login shows the name of the user for whom the card is provisioned, and provides a single text box in which the user can type the PIN associated with the card.



If the user is not enabled for the zone, or is not a valid Active Directory user at all, the smart card login screen is replaced by either a list of local users, or user name and password text entry fields.

The user will be successfully logged in if the following conditions are met:

- The user enters the correct PIN for the smart card.
- The card is trusted by the domain and has not been revoked. The card is checked locally first, online or offline, to ensure that the issuing certificate authority is trusted by the Red Hat Linux computer through the certification authority trust chain, which is set up when the computer joins the domain, and is periodically refreshed.

Checking is performed by the domain controller when the computer is online, and by a local service, based on cached CRLs, when the computer is offline. If the user is not connected to the network but has previously logged on — with a smart card or in some other way — the Delinea Agent gets the UPN from the card and looks up the user in the cached data.

If login fails, no feedback is provided to the user as to why the login is being denied. However, information is logged into various system log files, /var/log/system.log, /var/log/secure.log, and the Delinea log file (/var/log/centrifydc.log) if logging is enabled, that can help determine the reason for a denied login.

How Login Screen Appears for a Multi-User Card

When a user inserts a card that is provisioned for multiple users, the smart card login provides a **Username** box that allows the user to enter the name of the account to use.

When the system finds the user account in Active Directory, it prompts the user to enter the PIN for the card.

If the user is not enabled for the zone, or is not a valid Active Directory user at all, the smart card login dialog is replaced by the previous login screen, either a list of local users or username and password text entry fields.

The user will be successfully logged in if the following conditions are met:

- The user enters the correct PIN for the smart card.
- The card is trusted by the domain and has not been revoked. The card is checked locally first, online or offline, to ensure that the issuing certificate authority is trusted by the Red Hat Linux computer through the certification authority trust chain, which is set up when the computer joins the domain, and is periodically refreshed.

Checking is performed by the domain controller when the computer is online, and by a local service, based on cached CRLs, when the computer is offline. If the user is not connected to the network but has previously logged on — with a smart card or in some other way — the Delinea Agent gets the name from the log on screen and looks up the user in the cached data.

If login fails, no feedback is provided to the user as to why the login is being denied — as is the case when logging in with a password. Information is logged into various system log files that can help determine the reason for a denied login, /var/log/system.log, /var/log/secure.log, and the Delinea log file (/var/log/centrifydc.log) if logging is enabled.

Screen Saver Shows Password Not PIN Prompt

Most smart card users are allowed to log on with a smart card and PIN only — they cannot authenticate with a user name and password. However, it is possible to configure users for both smart card/PIN and user name/password authentication. Generally, this set up works seamlessly: the user either enters a user name and password at the log on prompt, or inserts a smart card and enters a PIN at the prompt.

However, for multi-user cards, it can be problematic when the screen locks and the card is in the reader. When a user attempts to unlock the screen, the system prompts for a password, not for a PIN, although the PIN is required because the card is in the reader. If the user is not aware that the card is still in the reader and enters his password multiple times, the card will lock once the limit for incorrect entries is reached.

What Happens After Login

In general the user experience is the same in both connected and disconnected modes, with the exception of single sign-on (SSO). Because the agent does not cache the smart card's PIN, single sign-on (SSO) is available for smart card authentication only while the computer is connected to the domain.

Of course, certain behaviors and system responses are specific to smart card login:

- If the user removes the smart card after logging on, the response of the system depends on whether the group policy "Lock smart card" screen is enabled in the domain. If it is, the screen locks. Otherwise, the screen does not lock and the user may continue working.

  Note: For a smart card that is provisioned for multiple users, if the screen locks, the system prompts for a Password, not for a PIN, when the user logs back in. However, the user must enter the PIN for the card, *not* the password, when logging back in.

- If the user inserts a smart card while the screen saver is active, the response depends on whether "Lock smart card screen" is enabled in the domain. If it is, the screen saver deactivates. If the policy is not enabled, the screen saver continues running until the user moves the mouse or touches a key.

Disabling Smart Card Support

If you want to disable smart card support, you must disable the group policies you configured to establish smart card authentication.

To Disable Smart Card Support by Using Group Policy

1. Edit the Group Policy object linked to the site, domain, or OU that includes Red Hat Linux computers.
2. Expand **Computer Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Enable smart card support**.
3. Select **Disabled** and click **OK**.

   When the policy takes effect, smart card strings are removed from /etc/pam.d/system-auth on Red Hat Enterprise Linux 5.6 and /etc/pam.d/smartcard-auth and /etc/pam.d/gnome-screensaver on Red Hat Enterprise Linux 6.0.
4. Expand **Computer Configuration** › **Policies** › **Centrify Settings** › **Linux Settings**, click **Security**, then double-click **Lock Smart Card screen for RHEL**.
5. Select **Disabled** and click **OK**.
6. To apply these group policies immediately to any computer, restart the computer or run the adgpupdate command on it.

   Otherwise, all affected computers will be updated automatically at the next group policy update interval. After computers are restarted or receive the policy updates, they are no longer enabled for smart card use.

To Disable Smart Card Support by Running sctool

1. Log on to a Red Hat computer with root privilege and open a terminal window.
2. Run the sctool utility with the --disable option:

   ```
   [root]$ sctool --disable
   ```
3. Repeat steps 1 and 2 for each computer on which to disable smart card authentication.

> **Note**: If you originally enabled smart card support through group policy by setting "Enable smart card support" you cannot disable it by using sctool --disable. Although this command will temporarily disable smart card support, it will be re-enabled by the policy at the next group policy update interval. To permanently disable smart card support, you must disable **Enable smart card support** as described in the previous procedure, To disable smart card support by using group policy.

Troubleshooting Smart Card Login

If you have problems with smart card login, Server Suiteprovides a command-line tool, sctool, that you can run to configure smart card login, as well as to provide diagnostic information. For example, you can run sctool with the following options:

- sctool --status to show whether smart card support is enabled.
- sctool --dump to display information about the smart card system setup as well as any smart cards that are attached to the computer.
- sctool --pkinit *userPrincipalName* to obtain Kerberos credentials on a single-user smart card for troubleshooting purposes.

  During login with a smart card, the agent calls sctool --pkinit to obtain Kerberos credentials from the smart card currently in the reader. Because this option simulates a good portion of the smart card login process, if you are having trouble logging in you can run sctool --pkinit to obtain useful troubleshooting information. If the command executes successfully, the name of the user will be displayed. If the command fails, you will receive an error message that may help you troubleshoot the issue.
- sctool --altpkinit *unixName* to obtain Kerberos credentials on a multi-user smart card for troubleshooting purposes.

  During login with a multi-user smart card, the agent calls sctool --altpkinit to obtain Kerberos credentials from the smart card currently in the reader (because the card is configured for multiple accounts, the user is prompted to provide a username, which the command uses to obtain the Kerberos credentials). Because this option simulates a good portion of the smart card login process, if you are having trouble logging in you can run sctool --altpkinit unixName to obtain useful troubleshooting information. If the command executes successfully, the name of the user will be displayed. If the command fails, you will receive an error message that may help you troubleshoot the issue.
- sctool --check-kdc-eku to enable checking of the KDC certificate for the Extended Key Usage (EKU) extension "Kerberos Authentication". Do not use this option if you have not updated your KDC to include the required EKU. Enable EKU checking after updating your KDC certificate.

  EKU checking is disabled by default.

  This parameter must be used with the -k (--pkinit) parameter or the -a (--altpkinit) parameter

For more information about using sctool, see the sctool man page.

Administration Guides

The following administrator guides are available

- Linux/Unix Administrator Guide
- Windows Administrator Guide
- macOS Administrator Guide
- Unix/Linux Quick Start
- MFA Guide
- Auto-Enrollment Guide

Refer to the following topics:

- Server Suite for Linux / Unix
- Managing Zones and Delegating Admin Tasks
- Managing Account Profiles and Identity Attributes
- Authorizing Basic Access
- Defining Rights to Use Commands
- Defining Rights to Use PAM Applications
- Using Secure Shell Sessions-based Rights
- Creating and Assigning Custom Role Definitions
- Working with Computer Roles
- Working with Managed Computers
- Importing sudoers Configuration Files
- Using Centrify OpenLDAP Proxy Service
- Using Workstation Mode and Auto Zone
- Troubleshooting Authentication and Authorization
- Using Centrify Commands for Administrative Tasks
- Using Python with Centrify Objects

Server Suite for Linux and UNIX

Server Suite is an IT management solution that provides key services for managing user and group profiles, role-based access rights, elevated privileges for administrative activity, and auditing-based regulatory compliance. These services can be used together or independently, depending on the requirements of your organization. The topics in this section introduce the key Server Suite that enable you to centrally manage Linux and UNIX computers. It includes an overview of how Centrify enables your organization to manage identity attributes, role-based access rights, and administrative activity through an integrated set of services.

Why Securing Access is Crucial

For most organizations, it is critical to control access to computer and application resources to prevent disruptions of service, data tampering, or security breaches. For many organizations, it is also critical to monitor and report on user activity to ensure regulatory compliance with government or industry standards. However, managing who has access to sensitive data, core business services, and the computers and applications that perform vital functions is especially difficult in data centers that include a mix of virtual and physical computers running different operating systems and platform versions.

Why Managing User Account Information Might be a Problem

In a cross-platform environment, you are likely to have multiple identity stores that might have overlapping or conflicting information about the user population. You might also have several different authentication methods—with varying degrees of security—that you are required to manage. For example, in a typical environment with a mix of Linux and UNIX computers, you might have to maintain any combination of the following authentication methods:

- Local configuration files on individual UNIX servers and workstations to identify local users and groups.
- NIS or NIS+ servers and maps to store account and network information for groups of UNIX servers and workstations.
- Kerberos realms and a Key Distribution Center to provide authentication for some users and services.
- Lightweight Directory Access Protocol services to support LDAP queries and responses.

Managing all of these services separately can be costly and inefficient. In addition, users who have access to more than one application or computer platform often have to remember multiple login accounts with conflicting user name or password policy requirements. Individual applications might also require the use of a specific authentication method. For example, a database application or a web service might require users to have a database- or application-specific account.

If you have an environment where user and group account information is stored in multiple locations rather than in a single repository, it is likely that you have overlapping, conflicting, or out-of-date information about who should have access to the computers in your organization. You might also be using less secure authentication and authorization services than required, if you are relying on local configuration files or NIS servers and maps. For example, if you are in an organization that is subject to regulatory compliance, an audit might require you to improve the security of the authentication and authorization services you use.

Why Managing Access and Privileges Might be a Problem

Most organizations require some groups of users to be allowed to use administrative accounts and passwords. For example, you might want to grant these permissions to allow some users to log on to computers that host administrative applications or data center services, but restrict access so that users can only log on when appropriate.

In many cases, the primary way you secure access to computers is by granting a limited number of users or groups root administrative privileges or configuring sudoers rights locally. These common practices leave computers vulnerable to insider threats and present a security risk that might be exploited by an external attack. As common as it is, granting administrative access rights is likely to violate the principal of least privilege, which is intended to minimize your exposure to these types of risks.

In other cases, users who need administrative privileges to perform specific tasks might use a shared administrator and service account password. However, shared passwords reduce accountability, leave computers vulnerable to insider threats, and are also often flagged by auditors as a security issue. If you are in an industry that has compliance requirements, shared passwords might present a significant business risk.

How Centrify can Reduce Security Risks

To reduce the overhead of managing account information and access rights across your organization, Centrify provides the following key features:

Secure Authentication and Identity Management

Centrify enables you to define and manage the identity attributes in user profiles, consolidate and simplify the management of account information, improve the security of authentication and directory services, and enforce consistent password and account policies.

Role-based Access Rights

Centrify enables you to define and manage access rights and role definitions, restrict which users can do what on specific sets of computers or during specific periods of time, and control and restrict access to administrative privileges.

Delegation of Authority

Centrify enables you to delegate administrative activity on a task-by-task basis. By delegating individual tasks to specific users or groups, you can establish a separation of duties at the level of granularity you require.

Auditing of Activity

Centrify enables you to collect and store an audit trail of user activity when and where you want it. With the auditing service, you can selectively capture and analyze only audit trail events or all user and computer activity.

These features can be used together or independently, depending on the type of licenses you purchase and the specific requirements of your organization. For example, some licenses for Server Suite might enable identity management, access control, and privilege management. Other licenses might enable auditing of user activity and reporting services.

How Zones Help you Organize Information

One of the most important aspects of managing computers with Centrify software is the ability to organize computers, users, groups, and other information about your organization into **Centrify zones**. A Centrify zone is a logical object that you create to organize computers, rights, roles, security policies, and other information into logical groups. These logical groups can be based on any organizing principle you find useful. For example, you can use zones to describe natural administrative boundaries within your organization, such as different lines of business, functional departments, or geographic locations. You can also use zones to isolate computers that share a common attribute, such the same operating system.

Zones provide the first level of refinement for access control, privilege management, and the delegation of administrative authority. For example, you can use zones to create logical groups of computers to achieve the following goals:

- Control who can log on to specific computers.
- Grant elevated rights or restrict what users can do on specific computers.
- Manage role definitions, including availability and auditing rules, and role assignments on specific computers.
- Delegate administrative tasks to implement “separation of duties” management policies.

You can also create zones in a hierarchical structure of parent and child zones to enable the inheritance of profile attributes, rights, roles, and role assignments from one zone to another or to restrict local or remote access to specific computers for specific users or groups.

Because zones enable you to grant specific rights to users in specific roles on specific computers, you can use zones as the first level of refinement for controlling who has access to which computers, where administrative privileges are granted, and when administrative privileges can be used.

You can also use zones to establish an appropriate separation of duties by delegating specific administrative tasks to specific users or groups on a zone-by-zone basis. With zones, administrators can be given the authority to manage a given set of computers and users without granting them permission to perform actions on computers in other zones or giving them access to other Active Directory objects.

Improving Security: Access and Privilege Management

Centrify provides its identity management, access control, and privilege management features for Linux and UNIX computers through a combination of features provided by Access Manager and by the Centrify Agent on the computers you want to manage.

You can install Access Manager and related management tools on one or more Windows computers. For example, the central console for performing most identity management, access control, and privilege management tasks is Access Manager. From Access Manager, you can perform all of the following common administrative tasks:

- Define and manage identity attributes for the Active Directory users who need access to Linux and UNIX computers.
- Import and migrate UNIX users, groups, and network information from local configuration files and NIS maps.
- Define and manage rights that allow users to run command-line programs, PAM applications, and secure shell operations.
- Select rights to create role-based access control role definitions and assign those roles to the appropriate users and groups.
- Delegate administrative tasks and control the specific permissions granted to users who are managing the computers in your organization.

For example, you can use Access Manager to delegate specific administrative tasks—such as the ability to add and remove users or assign roles—to a particular user or group. As an administrator, you can also use Access Manager to configure roles that have specific start and expiration dates or that limit the availability of a role to specific days of the week or hours of the day. You can use zones in combination with rights and roles to restrict or grant access to specific Linux and UNIX computers in your organization.

Through the use of zones and roles, Centrify provides granular control over **who** can do **what**, and control over **where** and **when** those users should be granted elevated privileges.

Consolidating User Account Information

Centrify enables you to consolidate all of your user and group account information in a single repository. By consolidating user account information, you can improve IT efficiency and overall operational security. For example, you can automate the provisioning of new accounts and the elimination of accounts that are no longer used without changes to your existing infrastructure or processes.

A single repository also enables you to establish consistent password policies for all of the computers you manage. For example, you can enforce consistent rules for password complexity and minimum length for all users on all computers. A single repository also benefits users, who only have to remember one password, regardless of the computer they use.

By using Centrify zones and override controls, you can migrate your entire user population without modifying any existing account attributes. For example, you can map multiple UNIX profiles with different identity attributes to a single user account, or resolve conflicts if the profiles for different users have the same identity attributes. This flexibility ensures that you can migrate legacy user accounts without changing any existing profile attributes, so that all of the existing directory and file ownership remains unchanged.

Over time, you can then continue to improve organizational security by eliminating legacy identity stores, directories, and databases, including all locally managed /etc/passwd files and local user accounts.

Defining Role-based Access Rights

Role-based access rights are more flexible than UNIX group membership rights and easier to define than user specifications in a sudoers configuration file. Role-based access rights can be narrowly applied or broadly inherited across any number of computers. You can restrict when role-based rights can be used by defining roles that are available only on certain days of the week or only during specific hours of the day. You can also make role assignments temporary by setting a date and time for the assignment to start or expire. For example, you might given the user Jonah elevated privileges to run administrative commands in the Backup Operators role for a period of two weeks while the primary backup administrator is on vacation.

Role-based access rights also prevent password sharing for privileged accounts, helping to ensure accountability. Users who need to run privileged commands can either temporarily elevate their privileges in an unrestricted login shell or be required to run the commands in a tightly controlled restricted shell without being prompted to provide the administrative password. All of their privileged or restricted shell activity can be traced to the account they used to log on.

Improving Accountability: Auditing User Activity

Centrify provides its auditing and analysis features through a combination of auditing components on Windows computers and the auditing features of the Centrify Agent on the computers you manage. The auditing service includes several components to support the multitier architecture of the auditing infrastructure. These components are installed on Windows computers to enable you to collect and store detailed information about user activity.

The central console for configuring the auditing infrastructure and managing audit-related features is Audit Manager. From Audit Manager, you can perform the following common administrative tasks:

- View the status of all audited computers and the other components of the auditing infrastructure.
- Manage the scope and security for auditing-related activity.
- Set permissions for the tasks granted to specific auditors.

There is also a separate Audit Analyzer console for searching and replaying captured activity.

Why auditing User Activity is Important

Just as it is important to protect assets and resources from unauthorized access, it is equally important to track what the users who have permission to access those resources have done. For the users who have privileged access to computers and applications with sensitive information, auditing helps ensure accountability and improve regulatory compliance. With the audit and monitoring service, you can capture detailed information about user activity and all of the events that occurred while a user was logged on to an audited computer.

If you choose to enable auditing on Linux or UNIX computers, the Centrify Agent on that computer starts recording user activity as soon as a user logs on. The agent continues recording until the user logs out or the computer is locked because of inactivity. The user activity captured includes an audit trail of the actions a user has taken and a keystroke record of the text that was entered (stdin) and the results that were displayed (stdout and stderr). The information recorded while a user is logged on—which is called a **session**—is collected as it happens, so you can monitor computers for suspicious activity or troubleshoot problems immediately after they occur.

Reviewing User Activity

When you audit user activity on a computer, the information is transferred to a Microsoft SQL Server database so that it is available for review and follow-up. Because sessions and audit trail events are stored in the database, you can create queries and reports to find information of interest. For example, you can search the stored user sessions to look for policy violations, command-line execution errors, or malicious activity that may have led to a service degradation or an outage.

In addition to saving the input and output recorded, sessions provide a summary of actions taken so that you can scan for potentially interesting or damaging actions without playing back a complete session. After you select a session of interest in Audit Analyzer, the console displays a list of commands in the order in which the user executed them. You can then select any command in the list to start viewing the session beginning with that action. For example, if the user ran a command that reports credit card information, you can scan the list of commands for the command that accesses credit card information and begin reviewing what happened in the session from that time on.

Using Access and Auditing Features Together

You can use access-related features and components without auditing if you aren't interested in collecting and storing information about session activities. You can also deploy auditing-related features and components without access control and privilege management features if you are only interested in auditing user activity on Linux and UNIX computers. However, you can recognize the most value from Server Suite by using all of the services as an integrated solution for managing elevated privileges and ensuring accountability and regulatory compliance across all platforms in your organization.

Enabling Access Control without Auditing on a Managed Computer

If you only enable access control features, the agent enforces the role-based privileges that enable users to log on, access PAM-based application, and run administrative or restricted shell commands. All of the role-based activity is traceable to the user's own account credentials. However, the audit trail of user activity is only recorded in the computer's local system log (syslog) facility. Information that is only stored in a computer's syslog facility can be more difficult to monitor and query than information stored in a central repository such as Microsoft SQL Server database.

Enabling Auditing without Access Control on a Managed Computer

If you only enable auditing, the agent captures detailed information about the command input and output in the login shell of the managed computer. All of the activity is stored in the Microsoft SQL Server database and available to you for queries and reports. However, there's no role-based enforcement of what activity is allowed on the audited computer.

Enabling Access Control and Auditing on a Managed Computer

If you use the infrastructure access management and auditing services together, you can define role-based access rights, restrict when and where roles are available, identify roles that should be audited, trace activity when roles with elevated permissions are selected and used, and play back session activity based on the criteria you choose.

By combining access management and auditing on the same computer, you can have an audit trail and, optionally, a video record of all actions performed with elevated privileges. For example, when you deploy access management, users must be assigned to a role with permission to log on. If they are allowed to log on and auditing is deployed, the agent begins auditing their activity. If a user accesses a PAM-based application or executes a privileged command, the action is recorded and can be traced back to the account used to log on.

The following illustration provides a simplified view of the architecture and flow of data when you deploy components for access control, privilege management, and auditing on a Linux or UNIX computer.

However, auditing requires database storage for the audited sessions audit trail events. Auditing also requires additional management of the network connections used to collect and transfer audit-related information from computers being audited to one or more databases where the sessions and audit trail events are stored. If you plan to use the infrastructure access management and auditing services together, you also need to decide which roles should require auditing and which features to enable on each computer you want to manage. In most cases, you choose whether to enable access control features, auditing features, or both feature sets when you install the agent on a computer.

Although it is not depicted in the illustration, you do not have to enable the auditing service to record audit trail events locally for successful or failed operations. By using the auditing service, however, you can store the audited sessions and audit trail events in a database and report on specific types of activity, such as the execution of privileged commands or access to applications and information that must be kept secure. With auditing enabled, the audit trail and the user activity are available for display, querying, and analysis from any computer where you install Audit Analyzer. Through rights and roles you can restrict access to sensitive information and control who can run commands with elevated privileges or perform administrative tasks. Through queries and reports, you can track all of the activity taking place—by user, computer, the time the activity took place, the role that was used, the command that was executed, or other criteria—to verify that only authorized users are performing authorized tasks and to investigate and correct any unauthorized access anywhere in your organization.

For complete information about setting up and managing an audit installation, see the *Auditing Administrator's Guide*.

Managing Zones and Delegating Administrative Tasks

Zones are the key component for organizing account profiles, identity attributes, role-based access rights, and role assignments. Zones also enable you to establish logical administrative boundaries and delegate specific administrative tasks to the appropriate users and groups for Linux and UNIX computers. This chapter describes the different types of zones and how to use Access Manager to create and manage zones, modify zone properties, and delegate administrative tasks to other users and groups in your organization.

Starting Access Manager for the First Time

The first time you start Access Manager, you can use the Setup Wizard to prepare the Active Directory forest with organizational units and containers for Centrify objects. From the Setup Wizard, you can create either the recommended deployment structure or a custom deployment structure and set all of the appropriate permissions for the objects automatically. If you skip this initial configuration, you can rerun the Setup Wizard at a later time or create organizational units and containers manually. At a minimum, however, you need to select a location in Active Directory for license keys and zones. For more information about the recommended organizational units and permissions, see the *Planning and Deployment Guide*.

What to do Before Updating Active Directory

Before you use Access Manager the first time, you should contact the Active Directory administrator to determine the appropriate location for the deployment structure and whether you have the appropriate rights for completing this task. The specific administrative rights required for this task depend on the policies of your organization and who has permission to create classStore and parent and child container objects in Active Directory.

Rights Required for this Task

If you don't have administrative rights to create container objects in Active Directory, a domain administrator in the forest root domain can run the Setup Wizard or manually create the container objects and set the rights on those objects to allow other users to complete the initial configuration without being members of an administrative group.

The following table describes the minimum rights that must be granted on manually created container objects for other users to successfully complete the configuration with the Setup Wizard.

Licenses container	Read all properties Create classStore objects Modify permissions	This object only
	Write Description property Write displayName property	This object and all child objects
	By default, all Authenticated Users have read and list contents permission for the Licenses container and all of its child objects.	
Zones container	Read all properties Create classStore objects Create Container objects	This object only
	Write displayName property	This object and all child objects

If you are a domain administrator and manually creating the container objects, you should add a security group for Zone Administrators to Active Directory. Set the following permissions on the parent Zones container to allow other users to manage zones.

Zones container	Read all properties Create Container objects Delete Container objects	This object only
	Write displayName property	This object and all child objects

Who Should Perform this Task

A Windows Active Directory administrator performs this task, depending on your organization's policies, by running the Setup Wizard or by manually creating container objects and notifying another user of the location of the container objects. The user who runs the Setup Wizard must be granted the rights required to create classStore objects.

How often you Should Perform this Task

In most organizations, you only do this once for an Active Directory forest. However, if you want to create more than one administrative boundary, you can create additional parent containers as needed.

Steps for Completing this Task

The following instructions illustrate how to run the Setup Wizard from Access Manager.

To update Active Directory using Access Manager:

1. Open Access Manager.
2. Verify the name of the domain controller and the user credentials for connecting to the forest, then click **OK**.
3. At the Welcome page, click **Next**.
4. Select **Use currently connected user credentials** to use your current log on account or select **Specify alternate user credentials** and type a user name and password, then click **Next**.
5. Select **Generate the Centrify recommended deployment structure** if you want to create all of the containers for the recommended deployment structure automatically.

   If you select this option, select whether you want to generate the default deployment structure or generate a custom structure, then click **Next**.

   - If you are generating the default structure, clicking Next enables you to select or create the location for the deployment structure in Active Directory. For example, if you want to create the top of the default deployment structure at the domain level, click **Next**, then click **Browse** to select the domain name. After you have selected a location, click **OK**. then click **Next** to create the deployment structure.
   - If you are generating a custom structure, clicking Next enables you to export the script that creates the default structure or run a script you have previously written.

     If you are generating a default or custom deployment structure, verify the successful execution of the script that creates the structure, then click **Next** to continue.
6. Verify the parent container for licenses is in the top-level Centrify container if you are using the default deployment structure or the container of your choice, then click **Next**.

   You can add other Licenses containers in other locations later using the Manage Licenses dialog box.
7. Review the permission requirements for the container, then click **Yes** to continue.
8. Type or copy and paste the license key you received, then click **Add**.

   If you received multiple license keys, add each key to the list of installed licenses, then click **Next**. If you received license keys in a text file, click **Import** to import the keys directly from the file instead of adding the keys individually, then click **Next**.
9. Verify the **Create default zone container** option is selected and the parent container for zones is in the top-level Centrify container or the container of your choice, then click **Next**.

   If you run the Setup Wizard at any time after the initial creation of the Zones container, this step displays the **Change default zone container** option and the current container location. Select this option and click Browse to change the default container for zones, then click **Next**.
10. If you are using the recommended deployment structure, click **Next** to continue.

    This option allows "self-service" join operations for computers in the Computers container. It is only applicable if you are not using the recommended deployment structure. If you want to support "self-service" join operations and are not using the recommended deployment structure, select **Grant computer accounts in the Computers container permission to update their own account information**, then click **Next**.
11. If you plan to use Access Manager to manage information stored in Active Directory and maintain data integrity, click **Next** to continue.

    You should select **Register administrative notification handler for Microsoft Active Directory Users and Computers snap-in** if you want to automatically maintain the integrity of the information in Centrify profiles.

    This option prevents Centrify profile information from being left "orphaned" when changes are made to Active Directory objects such as users and groups. This option is not selected by default because it requires you to be a member of Enterprise Admins or Domain Admins group for the forest root

domain.

12. Select **Activate Centrify profile property pages** if you want to be able to display Centrify profiles in any Active Directory context, then click **Next**.

    Setting this option ensures that displaying the properties for a user, group, or computer always displays the Centrify Profile tab regardless of how you navigate to the Properties dialog box.

13. Review and confirm your configuration settings, click **Next**, then click **Finish**.

What to Do Next

Create at least one parent zone.

Where you can Find Additional Information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- How zones help you organize information
- Improving security: access and privilege management
- Improving accountability: auditing user activity

Preparing to Create Zones

As discussed in How zones help you organize information, Centrify zones help you organize computers, users, groups, access rights and other information into logical groups similar to Active Directory organizational units or Network Information Service (NIS) domains. You have several options when choosing the type of zone to create, and the type of zone you select depends entirely on what your organization needs. The first decision to make is the type of zone to create:

- Hierarchical, which is the default and supports inheritance and overrides.
- Classic, which is backward-compatible to support older versions of the Centrify Agent.
- SFU, which supports the Microsoft Services for UNIX schema and rarely used.
- Auto Zone, which is a simplified "zone" for computers to join when you don't need any control over profiles, access rights, or roles and role assignments.

With the exception of SFU zones, you can mix and match any combination of zone types in the same Active Directory forest, as needed. For example, you can create one or more classic zones to support legacy agents, an Auto Zone for a group of computers that don't require the management of identity attributes or access rights, and hierarchical zones for the computers for which you want to actively manage access rights and privileges.

Creating Hierarchical Zones

Hierarchical zones enable you to establish parent-child zone relationships, allowing profile attributes, rights, role definitions, and role assignments to be inherited down the zone hierarchy. In most cases, you define information in a parent zone so that is available in one or more child zones, as needed. At any point in the zone hierarchy, you can choose to use or override information from a parent zone.

You should use hierarchical zones if your organization has any of the following requirements:

- You have existing user and group profiles that must be migrated with legacy identity attributes to maintain existing file ownership.
- You have user and group profiles that have conflicting identity attributes on different computers.
- You have users and groups that require different role-based access rights, privileges, and role assignments on different sets of computers.

If you are using hierarchical zones, you can use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for login access to Centrify-managed Linux and UNIX computers and for privileged command execution in hierarchical zones, classic zones, and Auto Zone. However, some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multifactor authentication, see "Preparing to use multi-factor authentication" and the *Multi-factor Authentication Quick Start Guide*.

Creating Classic Zones

Classic zones do not support inheritance or overrides and have other limitations in how they support role-based access rights. For example, in classic zones, authorization is disabled by default, and must be consciously enabled on a zone-by-zone basis before any role-based access rights or privileges can be configured or assigned.

You should only create new classic zones if your organization has any of the following requirements:

- You must support older versions of the Centrify Agent for *NIX.
- You have a user population with very few or no identity attribute conflicts.
- You have little or no need to centrally manage access rights and privileges.

If you are using classic zones, you cannot use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for access to Centrify-managed Linux and UNIX computers and for privileged command execution in classic zones. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multifactor authentication, see "Preparing to use multi-factor authentication" and the *Multi-factor Authentication Quick Start Guide*.

Creating an Auto Zone

Most organizations that deploy the Centrify Agent on Linux or UNIX computers have an existing user population to migrate to Active Directory, and hierarchical zones make the most sense. However, multiple zones are not required for all situations. You can greatly reduce the time required and complexity of your deployment if a single zone suits your organization's needs. This type of zone is created automatically when computers join the domain using the --workstation option.

An Auto Zone automatically enables all of the users and groups in an Active Directory forest to become valid users and groups on the Linux and UNIX computers that join the Auto Zone.Their profiles are generated automatically and there's no need to manage account profiles, access rights, privileges, or delegated administrative tasks.

You should only use the Auto Zone option if your organization meets the following requirements:

- You are not migrating an existing user population.
- You want to automatically generate profiles for all or most Active Directory users and groups without managing identity attributes.
- You don't want to configure and manage role-based access rights and privileges or role assignments.

If you are using an Auto Zone, you cannot use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for both licensed and Express agents to control access to Centrify-managed Linux and UNIX computers. For licensed agents, you can also require multi-factor authentication to run privileged commands in an Auto Zone. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multi-factor authentication, see the *Multi-factor Authentication Quick Start Guide*.

Creating a New Parent Zone

In most cases, you design a basic zone structure as part of the deployment process. After the initial deployment, you can create new hierarchical zones any time you have new administrative boundaries. For example, if you acquire another organization, add offices that are managed by a different group, or restructure the organization along different functional lines, you are likely to need new zones.

You can create as many parent zones as you need. You must create at least one new zone before you begin adding Linux and UNIX computers to the Active Directory domain, unless you are joining with the --workstation option.

What to do before creating a new parent zone

Before you can create parent zones, you must have installed Access Manager and run the Setup Wizard. You should also have a basic zone design that describes how you are organizing information, for example, whether you are using one top-level parent zone or more than one parent zone. You should also decide whether to create the new zone in the default Zones container object or in another container or organizational units within Active Directory. There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new zones, your user account must be a domain user with the following permissions:

Parent container for new zones, for example: domain/Acme/Zones	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: Create Container Objects Create Organizational Unit Objects **Note** Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create group objects Write Description property

Note: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who should perform this task

A Windows domain administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are fully deployed, you create new zones infrequently to address changes to your organization.

Steps for completing this task

The following instructions illustrate how to create a new parent zone using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a new parent zone using Access Manager:

1. Open Access Manager.
2. Select Zones, right-click, then click **Create New Zone**.
3. Type the zone name and, optionally, a longer description of the zone.

   In most cases, you should use the default parent container and container type that you created when you configured the Active Directory forest and use the default zone type, which creates the new parent zone as a hierarchical zone, then click **Next**.

The only reasons for changing the default settings would be if you want to:

- Create a zone in a new location to separate administrative activity for different groups of administrators.
- Create a zone as an organizational unit because you want to assign a Group Policy Object to the zone.
- Create a classic or SFU zone to support legacy Centrify Agents or to store data using the Microsoft Services for UNIX schema.

  For additional information about any field in the new zone wizard, you can press F1 to view the context-sensitive help.

4. Review information about the zone you are creating, then click **Finish**.

What to do next

After you create a new parent zone, you might want to create its child zones.

Where you can find additional information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- How zones help you organize information
- Preparing to create zones

Creating Child Zones

The primary reason for creating child zones is to inherit profile attributes, role definitions, and role assignments from a parent zone. You can then use the child zone to override the specific profile attributes that might be different on a given set of Linux and UNIX computers than you have defined in the parent zone. Less often, you might want to use a child zone to override specific access rights, role definitions, or roles assignments that you have made in a parent zone. For example, if you have created a role definitions that allows a user to run a specific application with administrative privileges in a parent zone, you can use child zones to limit the scope of that right to specific subsets of computers.

What to do before creating child zones

Before you create child zones, you must have installed Access Manager, run the Setup Wizard to create the Zones container, and created at least one parent zone. You should also have a basic zone design that describes the zone hierarchy for the child zone. There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new child zones, your user account must be a domain user with the following permissions:

Container for the parent zone, for example if the parent zone is berlin: domain/MyOU/Zones/berlin	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: Create Container Objects Create Organizational Unit Objects **Note** Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create group objects Write Description property These permissions are only needed if you are supporting “agentless” authentication in the new zone.

Note: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who should perform this task

A Windows administrator performs this task, depending on your organization’s policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are fully deployed, you create new child zones infrequently to address changes to the scope of ownership and administrative tasks.

Steps for completing this task

The following instructions illustrate how to create a new child zone using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer’s Kit*, or in community forums on the Centrify website.

To create a new child zone using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new child zone.
3. Right-click, then click **Create Child Zone**.
4. Type the zone name and, optionally, a longer description of the zone.

   Because this is a child zone, you should use the default parent container and container type, then click **Next**.
5. Review information about the child zone, then click **Finish**.

Opening and Closing Zones

Because properties and objects are organized into zones, you must open a zone to work with its contents. If you open a parent zone, its child zones are also available for you to use by default. If you open a child zone, you can choose whether to open its parent zone.

To open an individual parent or child zone:

1. Open Access Manager.
2. Select Zones, right-click, then click **Open Zone**.
3. Type all or part of the name of the zone you want to open, then click **Find Now**.
4. Select the zone to open from the list of results, then click **OK**. You can use the **CTRL** and **SHIFT** keys to select multiple zones.

Loading all zones

As an alternative to opening individual or parent and child zones manually, you can automatically load all zones in a forest or all zones in a specific container at startup time. If you choose to load all zones, you cannot manually close zones.

To load all zones automatically:

1. Open Access Manager.
2. Select Access Manager, right-click, then click **Options**.
3. On the **Filter Settings** tab, select **Load all zones**, then select **connected forest** to automatically load all zones in the forest or click **Browse** to navigate to specific container.

You should not select the Load all zones option if you want to manually open and close zones for performance reasons.

Closing individual zones

After you open a zone, it stays open during your current sessions unless you close it. If you have a large number of zones, however, you should close any zones you aren't actively working with for better performance.

To close an open zone:

1. Open Access Manager.
2. Expand Zones and select an open parent zone, right-click, then click **Close**.
3. Click **Yes** to confirm that you want to close the zone.

Delegating administrative tasks

If you have created at least one zone, you can give other users and groups permission to perform specific types of administrative tasks within that zone. For example, assume you have created a new zone called Finance and you want to give the users who access computers in this zone the permissions required to perform certain kinds of tasks based on their role. You can accomplish this goal by selecting a group or users, then assigning that group or user one or more tasks. For example, in the Finance zone, you might want to delegate administrative tasks like this:

- The members of the Active Directory group FinanceITStaff are allowed to perform all administrative tasks in the Finance zone.
- The members of the Active Directory group FinanceManagers are allowed to add, modify, and remove user and group profiles in the Finance zone.
- The members of the Active Directory group FinanceUsers are allowed to join computers to the Finance zone, but perform no other tasks.
- The Active Directory users jason.ellison and noah.stone are granted permission to manage role assignments in the Finance zone.

In most cases, each zone should have at least one Active Directory group that can be delegated to perform all administrative tasks, so that members of that group can manage their own zone. You are not required to create or use a zone administrator group for every zone. However, assigning the management of each zone to a specific user or group creates a natural separation of duties for administrative tasks.

If you delegate control for individual tasks—for example, by assigning only the join computers task to one group and only the add and remove users tasks to

another—you should ensure the members of each group know the tasks they are assigned.

You can delegate administrative tasks for parent zones, for child zones, and for individual computers. Because computer-level overrides are essentially single computer zones, you can assign administrative tasks to users and groups at the computer level.

What to do before delegating administrative tasks

Before you delegate administrative tasks for a zone, you must have created at least one zone. For each zone you create, you should also identify at least one user or group that can be delegated to perform all administrative tasks. For example, if you have a Finance zone, you might want to create a Finance Admins group in Active Directory, then delegate **All** tasks to that group so that members of that group can manage their own zone.

There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups.

For information about the permissions set when you select different administrative tasks in the Zone Delegation Wizard, see the *Planning and Deployment Guide*.

Who should perform this task

The domain administrator who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. Only the account used to create a zone has full control over the zone's properties and permission to delegate administrative tasks to other users. The user who creates a zone is also the only user who can add NIS maps to the zone. The right to create NIS maps is exclusive to the creator of a zone because it requires permission to create containers in Active Directory. The zone creator can, however, grant other users permission to add, remove, or modify NIS map entries.

How often you should perform this task

In most organizations, you delegate administrative tasks any time you create a new zone. You also might change the delegation to change the either tasks assigned or the users and groups that have been assigned specific tasks periodically to address changes to your organization. For example, if an existing zone administrator takes over new responsibilities or leaves the organization, you might need to delegate additional tasks or select a different user or group to perform administrative tasks.

Steps for completing this task

The following instructions illustrate how to delegate zone administration tasks to a user, group, or computer using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To delegate administrative tasks to specific users and groups in a zone:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name for which you want to delegate administrative tasks.
3. Right-click, then click **Delegate Zone Control**.
4. Click **Add** to find the users, groups, or computer accounts to which you want to delegate specific tasks.
5. Select the type of account—**User**, **Group**, or **Computer**—to search for, type all or part of the account name, then click **Find Now**.
6. Select one or more accounts from the list of results, then click **OK**.
7. When you are finished adding users and groups to which you want to assign administrative tasks, click **Next**.
8. Select the tasks you want to delegate to the user or group, then click **Next**.

   For example, if you want all of the members of the group you selected in the previous step to be able perform all administrative tasks for a zone, check the **All** task. To restrict the administrative tasks a user or group can perform, select only those specific tasks.

   restrict the administrative tasks

9. If you are delegating the task of joining computers to a zone, you can specify the scope of computers you can join to the zone; you pick a container in Active Directory to grant access to.

   If you leave the scope blank, the scope is the domain root. Be aware that the postalAddress field is used for information about joining computers to a zone; if you lookup the permissions for people you've delegated the task of joining computers to a zone, they'll have permissions to the postalAddress field for the affected computers.

10. Review your selections, then click **Finish**.

If you have delegate administrative tasks to one or more groups that have members logged on, you should notify the group members to log out and log back on before they attempt to perform the administrative tasks assigned to the group.

Changing Zone Properties

After you create a zone, you can change its zone properties at any time. For example, if you want to change the parent zone for a child zone, you can do so by modifying the child zone's properties. Depending on whether you are viewing a classic, hierarchical, or SFU zone and the components you have installed, you might see and be able to set different zone properties.

To display the properties for a zone:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.

   If the zone you have selected is a hierarchical zone, the properties are organized on the following tabs.

General	View and set general information about the selected zone, including the location of the zone in Active Directory, the zone type, and the zone description. For additional details about general properties, see the following topics:
	Changing the zone description
	Changing the parent zone or location of a zone
	Setting the master domain controller for a zone
	Selecting a license container for a zone
	Adding support for agentless clients
	Setting custom permissions for a zone
Platform	View and set the identity platform instance to use for the selected zone. For additional details about setting identity platform properties, see the following topic: Selecting a identity platform instance for a zone
User Defaults	Set default values for user profile attributes in the selected zone. For additional details about user default properties, see the following topic: Setting user defaults
Group Defaults	Set default values for group profile attributes in the selected zone. For additional details about group default properties, see the following topic: Setting group defaults
Variables	Add or edit user-defined variables or override the default values of predefined variables in the selected zone. For additional details about zone variables, see the following topic: Configuring variables for a zone
Provisioning	Configure automated provisioning for user and group profiles if you have the Zone Provisioning Agent installed on the local computer. For additional details about provisioning properties, see the following topic: Configuring automated provisioning The Provisioning tab is only displayed if the Zone Provisioning Agent is installed. For detailed information about configuring automated provisioning, see the *Planning and Deployment Guide*.

Changing the zone description

You can set or change the optional description for a zone at any time. For example, if you didn't specify a description when you created the zone or if there have been changes in your organization that warrant a change in the description of a zone, you can modify the Description field to make the change.

To change the zone description

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Type a description for the zone in the Description field, then click **OK**.

Changing the parent zone or location of a zone

From Access Manager, you can make any existing hierarchical zone the child of another zone or make any child zone a new parent zone by dragging and dropping the zone into a new location or by changing the Parent zone field on the zone's General properties tab.

Selecting the default location when moving a zone

If you make changes to the zone hierarchy, Access Manager prompts you to specify the new Active Directory location for the zone. In most cases, you should accept the default location for the zone you are moving. The default Active Directory location will be either:

- The **new parent zone** container if you are moving a child zone from one parent to another or if you moving a parent zone to become a child zone.
- The default **Zones** container you created the first time you started Access Manager if you are making a child zone a new top-level parent zone.

You are not required to accept the default Active Directory location when changing the zone hierarchy. If you select a different Active Directory location for the zone, however, you should note the location and whether the zone you are moving is now a parent or a child zone. If the zone structure displayed in Access Manager is different from the zone container structure you are using in Active Directory, you might find unexpected problems with inheritance and overrides, with modifying zone properties, or with deleting zones.

Moving a zone without changing its Active Directory location

When you are prompted to specify the Active Directory location for a zone you are moving, you have the option to select **No** and leave the current Active Directory location unchanged. If you change the parent zone without changing the Active Directory location for a zone, you should note that the location does not reflect the zone hierarchy. In rare cases, you might find it useful to leave the Active Directory location unchanged but doing so might make it more difficult to locate the zone object at a later time.

Restarting the agent after moving a zone

If you change the location for a zone in Active Directory, you must restart the Centrify Agent for *NIX on the computers in that zone so that they recognize the new zone location.

After you move the ZoneName object to a new parent container or organizational unit, run the following command to restart the Centrify Agent for *NIX on the computers in the zone:

/usr/share/centrifydc/bin/centrifydc restart

To move a zone to a new parent by changing properties

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. For the Parent zone field, click **Browse** to find and select the zone to use as the parent, then click **OK**.
5. Click **OK** to save the new zone properties.
6. In the Move Zone dialog, verify the location selected for the **Yes, move to** option to accept the default location, then click **OK**.

   In rare cases, you might want to click **Browse** to select a different Active Directory location for the zone you are moving, or select **No**, then click **OK** to keep the zone in its original location.

Setting the master domain controller for a zone

In most cases, computers connect to the first available Active Directory domain controller and it is not necessary to specify the master domain controller to use for a zone. In some cases, however, you might want to identify a specific domain controller to use for a zone to prevent connections from other domain controllers from adding or removing users and groups in that zone.

To prevent connections from other domain controllers, you can set the Master domain controller field to the fully-qualified name of the domain controller you want to use. After you identify a master domain controller, administrators who connect to the zone using any other domain controller will not be able to make changes to the zone.

If you have multiple administrators managing any zones, you should notify them before setting or changing the master domain controller. You should also make this change while all other administrators are logged off. Depending how long it takes for replication to complete for all of the domain controllers in the Active Directory forest, you might want to schedule this change for a time when no administrators need access to zone information.

To change the master domain controller

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to change the master domain controller.

   You can use Shift-Click or Ctrl-Click to select multiple zone names.
3. Right-click, then click **Change Master Domain Controller**.
4. Type the fully-qualified domain name for the new domain controller, then click **OK**.
5. Click **Yes** to confirm that you want to change the master domain controller for the zone.

You should avoid changing from one master domain controller to another, if possible. Changing the master domain controller requires you to wait for replication to complete to see up-to-date zone information or modify information in the selected zone. In some cases, however, changing the master domain controller might be unavoidable. For example, if there are zones connecting to a master domain controller that has a hardware failure or must be taken offline for maintenance, you will need to configure a new master domain controller for the zones to use.

If you change the master domain controller, you should run the Analyze command afterwards to check the Active Directory forest and verify that no duplicate UIDs or GIDs have been introduced.

Selecting a license container for a zone

By default, zones are configured to use any available license container in the forest. In most cases, the container used is the default **Licenses** container you created the first time you started Access Manager. If you have more than one Licenses container, you might want to select a specific license container for a set of computers in the one zone and a different license container for a set of computers in another zone. For example, you might want to select separate Licenses containers for the zones associated with two different business units.

To use a specific license container for a zone, you can type the path to a new container object in the License container field.

To use a specific license container for a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Select a specific license container from the list of available License container, then click **OK**.

For more information about licenses keys and using multiple license containers, see the *License Management Administrator's Guide*.

Adding support for agentless clients

If you are using the Centrify Network Information Service (adnisd) on a managed computer to respond to NIS client requests from computers where the Centrify Agent cannot be installed, you can configure one or more zones to act as the NIS domain for those client requests.

To add support for agentless NIS clients in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Select the **Support agentless client** option.
5. Select the Active Directory attribute you want to use to store the password hash and verify the zone name is the NIS domain name you want to use or type a new name, then click **OK**.

For more information about installing and using the Centrify Network Information Service (adnisd) to respond to NIS client requests and configuring agentless clients, see the Network Information Service Administrator's Guide.

Setting custom permissions for a zone

For convenience, you can access Permissions for a zone directly from the zone properties General tab. You can then allow or deny basic permissions—such as Read and Write permissions—to specific users and groups or click **Advanced** to set more granular permissions on a zone.

Selecting a identity platform instance for a zone

In most cases, the identity platform instance property is set automatically when you register a connector for Privileged Access Service. If you have access to more than one identity platform instance—for example, if you have more than one customer identifier, you can select the URL for a specific instance from the zone properties.

To select a identity platform instance for a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Platform** tab.
5. Verify the identity platform instance URL is the customer-specific URL you want to use, or click **Browse** to select the URL for a different customer-specific identity platform instance.

   Child zones inherit the identity platform instance property from their parent zone. If you are viewing properties for a child zone, you can select **Override trusted identity platform instance** then click **Browse** to select a different identity platform instance for the child zone.

   For details about installing and configuring a connector, see "Preparing to use multi-factor authentication."
6. Click **OK** to confirm the identity platform instance selected.

Configuring default values for a zone

You can configure default settings for user and group profiles that are added to the zone. The user and group defaults you configure can include predefined variables that populate the user or group profile by using Active Directory attributes or settings configured on individual managed computers.

By specifying user default and group default settings, you can simplify the process of adding user and group profiles to child zones. For example, you can define a default user profile that uses the sAMAccountName attribute for a user's UNIX login name. All users who are added to the zone are then automatically assigned a UNIX login name based on their sAMAccountName. If you define the default attributes in a parent zone, they can also be inherited in all of the child zones under that parent and only overridden where other values are explicitly required.

Setting user defaults

When you create a zone, it includes a default set of user profile attributes. In most cases, there's no need to modify any of the default settings unless you want to define partial profiles in a parent zone that will be manually completed in child zones. For example, the default setting for the numeric user identifier (UID) is an automatically generated UID based on the user's globally unique security identifier (SID). This setting ensures all users who are added to the zone are assigned a unique UID for the entire forest.

If you define a default value for any user profile attribute, that value is used to populate the user profile displayed when you add users to the selected zone. When you add a user to the zone, you can accept the default profile attributes or override any of the default attributes displayed.

To view or modify the default user profile in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **User Defaults** tab.
5. Review the default settings and modify any of the defaults, if needed.

   For most organizations, the default settings are appropriate. For example, the Active Directory sAMAccountName attribute most closely resembles the most common format for the UNIX login name and an automatically generated UID ensures that all new users have a unique UID in the forest. For more information about the attribute fields or the default values, press F1 to view the context-sensitive help.
6. Click **OK**.

For more information about using default values, see Creating user profiles for Active Directory users. For more information about using predefined or custom variables in user profiles, see Setting runtime variables in user profiles.

Setting group defaults

When you create a zone, it includes a default set of group profile attributes. In most cases, there's no need to modify the default settings for groups unless you are manually assigning numeric group identifiers (GID) or using the Apple algorithm for generating the GID.

If you define a default value for a group attribute, that value is used to populate the group profile displayed when you add groups to the selected zone. When you add a group to the zone, you can accept the default profile attributes or override any of the default attributes displayed.

To view or modify the default group profile in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Group Defaults** tab.
5. Review the default settings and modify any of the defaults, if needed.

   For most organizations, the default settings are appropriate. For example, the Active Directory sAMAccountName attribute most closely resembles the most common format for the group name and an automatically generated GID ensures that all new group have a unique GID in the forest. For more information about the attribute fields or the default values, press F1 to view the context-sensitive help.
6. Click **OK**.

For more information about using default values, see Creating group profiles for Active Directory groups. For more information about using predefined or custom variables in user profiles, see Setting runtime variables in user profiles.

Configuring variables for a zone

Predefined and custom variables enable you to generate user profiles and group profiles using Active Directory properties or properties defined on managed computers.

You can add custom runtime variables, or override the definition for predefined variables, in a zone by modifying the zone properties. Runtime variables are resolved by the agent when a computer joins a zone. The default user profile settings use predefined runtime variables in place of specific values for the GECOS, Home directory, and Shell attributes.

Zone variables and their definitions are inherited down the zone hierarchy, and can be overridden in a child zone or on individual computers. You can also use configuration parameters to control the value for any variables locally on particular computers. If a value is set in the configuration file, it overrides any values

that you set for the zone.

Adding custom runtime variable

In most cases, you don't need to add custom variables to a zone. However, if you have modified the Active Directory schema or want to use custom attributes in user or group profiles, you can add custom variables to the zone to accommodate your changes.

To add a custom variable to a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Variables** tab.
5. Click **Add**.
6. Type a variable name and a value, then click **OK**.

   For example, you might want to define a custom variable named gecos and set its value to a static string, such as Engineering-Nova Scotia-Q22, for a zone.

   Similarly, you might want to add custom variables for different operating systems you support, such as mac-home or aix-shell for a zone that includes computers with different operating systems. For example, if a zone includes Linux, AIX, and Mac OS X computers, you might have a default profile that uses the predefined variables, but a subset of accounts that use the mac-home or aix-shell custom variables.
7. Click **OK** to save the properties.

Modifying predefined variable values

In most cases, you don't need to override predefined variable values for a zone. However, if you have created different zones for different operating systems, you might find it useful to modify predefined variable values for those zones to address different operating system requirements.

To modify a predefined variable value in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Variables** tab.
5. Click **Add**.
6. Type the name of a predefined variable and a value, then click **OK**.

   For example, you might want to change the predefined variable named home and set its value to an appropriate home directory for the zone, such as /export/home for a zone where all of the computers are Solaris computers, or /Users for a zone with only Mac OS X computers. Similarly, you might want to change the predefined variable shell to set its value to /usr/bin/ksh for a zone with IBM AIX computers.
7. Click **OK** to save the properties.

Editing or removing variables

After you have added custom variables or modified predefined variable values in a zone, you can later select those variables to edit or remove them.

Configuring automated provisioning

The Centrify Zone Provisioning Agent is a separate service that enables automated provisioning and de-provisioning of user and group accounts on a zone-by-zone basis. You can configure the Zone Provisioning Agent to monitor specific Active Directory groups for a zone. If you add or remove Active Directory users or groups in the monitored groups, the Zone Provisioning Agent automatically adds or removes the corresponding user or group profiles in the zone. If

you have the Centrify Zone Provisioning Agent installed, you can use the zone properties Provisioning tab to do the following:

- Enable provisioning for users, groups, or both.
- Specify the Active Directory group to base provisioning on.
- Select the method for automatically generating profile attributes for users, groups, or both.

For more detailed information about automated provisioning and using the Zone Provisioning Agent, see the *Planning and Deployment Guide*. For more information about the attribute fields or the options for generating profile attributes, press F1 to view the contextsensitive help.

Renaming a Zone

You can rename a zone at any time. For example, if your organization changes how business units are aligned, moves to a new location, or merges with another organization, you might want to update zone names and descriptions to reflect these changes. You might also want to rename zones if your initial deployment did not use a naming convention for new zones, and you want to implement one after you have agents deployed.

What to do before renaming a zone

Before you rename zones, you might want to define and document a naming convention to use for future zones or the reasons for changing the zone name. You should also identify the computers in the zone to be renamed. You must restart the agent on those computers for the new zone name to be recognized. There are no other prerequisites for performing this task.

Rights required for this task

To rename a zone, your user account must be set with the following permissions:

Parent container for an individual zone For example, a ZoneName container object, such as: domain/Zones/arcade	Write Description Write name Write Name These are the minimum permissions required to rename a zone and not allow a user or group to modify any other zone properties. You can set permissions manually, or automatically grant these and other permissions to specific users or groups by selecting the **Change zone properties** task in the Zone Delegation Wizard.

Who should perform this task

A Windows administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are deployed, you rename zones only when you need to address organizational changes or to implement or improve the naming conventions you use.

Steps for completing this task

The following instructions illustrate how to rename a zone using Access Manager.

To rename a zone using Access Manager:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand any child zones in the zone hierarchy until you see the specific zone you want to modify.
3. Select the zone to change, right-click, then click **Rename**.
4. Type the new name and, if needed, any changes to the zone description.
5. Restart all of the Centrify UNIX agents on the computers in the zone you have renamed.

   You do not have to leave and rejoin after changing a zone name. However, you must restart the agent for the name change to take effect on a managed computer.centrifydc In a terminal window on each managed computer, run the following command:

   /usr/share/centrifydc/bin/centrifydc restart
6. You can verify the updated zone name on a local computer by using the adinfo command, which includes the joined zone name in its output.

Adding Computers to a Zone

You can only join a domain by creating a computer account that is either a "zone computer" profile or a "workstation" account that uses Auto Zone. Depending on the tool and operating system you prefer to use, there are several ways you can add a computer account to a zone. For example, if you prefer to create the "zone computer" account from a Linux or UNIX computer:

- You can run the adjoin command interactively or in a script and specify the zone as a command line option while joining the Active Directory domain.
- You can use ADEdit commands interactively or in a script to add a computer account to a zone before joining the domain.

If you prefer to create the "zone computer" account from a Windows computer:

- You can prepare a computer account in Access Manager before joining the domain.
- You can use the Centrify Access Module for PowerShell cmdlets interactively or in a script to add a computer account to a zone.

Precreating computer accounts using ADEdit or the Centrify Access Module for PowerShell cmdlets is particularly useful if you want to join multiple computers with minimal command line options and if you want to allow the computer account to be used to perform a "self-service" join. For more information about preparing a computer account before joining a domain, see Preparing computer accounts before joining.

For more information about specifying the zone, joining the domain, and modifying computer properties, see Working with managed computers. For information about using Auto Zone, see Using workstation mode and Auto Zone.

Managing Licenses

The first time you start Access Manager, you are prompted to create a Licenses container and add or import license keys. You can also add and remove license containers and keys after the initial configuration.

To modify license information

1. Open Access Manager
2. Select Access Manager, right-click, then select **Manage Licenses**.
3. Click Add to add a new license container or license key.
4. Select an existing license container or license key, then click **Remove** to remove that container or key.

For details about licensing, including how to request new license keys after deployment, check license usage and compliance, and how license counts are determined, see the *License Management Administrator's Guide*.

Reporting Zone Information

You can access legacy reports from within Access Manager by selecting Access Manager, right-clicking, then selecting **Report Center**. You can also use command-line programs, PowerShell scripts, or ADEdit scripts to report zone information. In most cases, however, you should install and configure Report Services to generate and access reports about the Active Directory domain and your zones.

For details about installing and configuring Report Services, and how to customize and access the reports that generated, see the *Report Administrator's Guide*.

Migrating from Classic to Hierarchical Zones

Classic zones are primarily intended for backward compatibility with older versions of the Centrify Agent. If you upgrade the agent to version 5.x or later, you can migrate any or all of your classic zone information into one or more hierarchical zones.

Migrating a classic zone to a hierarchical zone is a multi-step process that requires some initial planning. For example, the first step in the migration changes the zone type but does not change any existing zone information, including the computer accounts that are joined to the zone. To take full advantage of the hierarchical zone after migration, however, it is likely that you will need to modify some of your existing zone information and move computer accounts into different zones.

Preparing for migration

To prepare for the migration of any classic zones, you should first review the existing zone information for "dominant" user and group profiles—that is, profiles with attributes that are common to multiple classic zones. Dominant profiles will help you to identify one or more classic zones that you can use as potential parent zones. A parent zone provides a baseline for the user and group profiles that can be inherited in child zones. The parent zone also enables you to manage rights and role definitions that can be inherited in the child zones you create. If you are able to identify dominant profiles, most of your classic zones will become child zones that inherit information from the parent zone, with specific attribute overrides on a zone-by-zone or computer-by-computer basis, as necessary.

To illustrate how you should analyze your existing environment, assume you have several classic zones to address different profile requirements on different computers, but only two administrative groups that have different policies and procedures for adding users or granting privileges. In this scenario, you might create two parent zones—one for each administrative team—and use child zones or computer overrides to address specific profile attribute differences. If your organization has a single account fulfillment desk that handles all provisioning and access privileges, you might create a single parent zone for managing all or most user and group profiles, then use child zones to manage more granular account privileges.

If you have a "master" classic zone where the most commonly-used profile attributes for most of your users and groups are defined, that zone is a likely candidate to become a hierarchical parent zone. If none of your existing classic zones is suitable to become a parent zone, you should create a new parent zone as described in Creating a new parent zone. The parent zone must exist before you can use the migration utility.

Verifying you have upgraded Access Manager

You can use Access Manager to view and manage any combination of zones. However, the console must be version5.x or later to work with hierarchical zones. You can check the version of the console you have installed by opening Access Manager, clicking Help, then selecting About Access Manager.

Verifying you have upgraded UNIX agents

The migration utility is a command-line program installed with the Centrify Agent for *NIX. You must upgrade the agent to version 5.1, or later, on at least one UNIX computer to do any migration. You can verify the agent version by running the adinfo command with the --version (-v) option.

What the migration utility does

After you have identified at least one classic zone as potential parent zones, you can use the migration utility to convert the classic zone into a hierarchical parent zone. After you make the classic zone a hierarchical zone, you can run the migration utility to make other classic zones into child zones of the parent zone.

During the migration, all of the user and group profiles in the source zone are copied to the specified parent zone. If identical profiles exist in multiple classic zones, the identical profiles become a single profile in the parent zone. If there are user or group profiles in multiple zones with different attribute values—for example, a UID of 10001 in one classic zone, but 10003 in another classic zone, the migration utility creates a single profile for the user in the parent zone, and creates a profile override with the distinct attribute values in each target child zone. Each child zone inherits the base profile from the parent zone but applies the overrides for any attributes that are different in different zones.

The migration utility copies everything else—including rights, roles, role assignments, groups, and NIS maps—into the new child zone for each classic zone being migrated.

Using the migration utility

Centrify provides the command-line program admigrate to simplify the process of migrating profiles, rights, roles, role assignments, and NIS maps from a classic zone to a hierarchical zone.

The admigrate program is installed by default in the following directory:

/usr/share/centrifydc/adedit/admigrate

Note that the first zone you migrate becomes the primary source of profile information for the other zones you migrate. You should start with the zone that it contains the most consistent profile attributes.

Note: Admigrate does not migrate classic SFU zones (Ref: CS-28289a) nor zone delegation rights (Ref: IN-90002).

To migrate zone information from a classic to hierarchical zone:

1. Log on to a Linux or UNIX computer running adclient and open a terminal window.
2. Open a text editor to create a file with bind information for each domain to which admigrate must connect.

   Specify the Active Directory credentials for an account with permission to create child zones, rights, roles, user profiles, and group profiles in the parent zone with one line per domain in the format:

   bind domain_account_password

   For example, create a file named migrate.conf with information similar to the following:

   bind finance.acme.com administrator {1234abcepassword}
   bind eng.acme.com engadmin {1234abcepassword}
3. Save and close the file.
4. Run the admigrate command.

   admigrate -in classicZone -z targetZone -hz parentZone -config configFile

classicZone	The distinguished name of the classic zone to migrate. For example: “cn=finance,cn=zones,ou=unix,dc=acme,dc=com”
targetZone	The distinguished name of the new zone. It can be the same as the existing classic zone name, however the new zone will be a child zone of the specified parent zone, so the distinguished name is different. For example: “cn=finance,cn=global,cn=zones,ou=unix,dc=acme,dc=com”
parentZone	The parent zone for the migration. The specified zone must be an existing zone. The target zone becomes a child zone of this zone. You can run admigrate multiple times and specify the same parent zone and different source and target zones each time to migrate multiple zones to different child zones of this parent. For example: “cn=global,cn=zones,ou=unix,dc=acme,dc=com”
configFile	The configuration file to use with the migration. The configuration file is primarily useful to specify bind information if you are migrating zones from domains that are different from the target zone's domain. The file is a simple text file, for example: -config admigrate.txt

For more information about other options you can use when running admigrate, see the man page for admigrate.

The first time you run admigrate, the command copies all of the user profiles from the source zone to the parent zone. Everything else defined in the source zone—including groups, rights, role definitions, role assignments, and NIS maps—is copied from the source zone to a new target child zone.

1. Repeat Step 4 for each classic zone you want to migrate as a child of the parent zone.

Sample migration

To illustrate how to use the admigrate command, assume you are migrating two classic zones—finance and engineering—into a new empty parent zone named global. For this example, the distinguished name of the classic finance zone (the source zone) is this:

“cn=finance,cn=zones,ou=unix,dc=test,dc=org”

After migration, the distinguished name of the finance child zone (the target zone) is this:

“cn=finance,cn=global,cn=zones,ou=unix,dc=test,dc=org”

To migrate the classic finance zone, you would run a command similar to the following:

```
/usr/share/centrifydc/adedit/admigrate \
-in "cn=finance,cn=zones,ou=unix,dc=test,dc=org" \
-z "cn=finance,cn=global,cn=zones,ou=unix,dc=test,dc=org" \
-hz "cn=global,cn=zones,ou=unix,dc=test,dc=org" \
-config ~/migrate.conf \
-v > migrate_finance.txt
```

In this example, the target zone name is the same as that of the input classic zone, except its distinguished name is different because it is a child zone of the *global* zone. The -config parameter specifies the file that contains bind information, in this cases ~/migrate.conf. The -v option directs verbose output to a text file.

You would then run admigrate for the next zone to migrate. For example:

```
/usr/share/centrifydc/adedit/admigrate \
-in "cn=engineering,cn=zones,ou=unix,dc=test,dc=org" \
-z "cn=engineering,cn=global,cn=zones,ou=unix,dc=test,dc=org" \
-hz "cn=global,cn=zones,ou=unix,dc=test,dc=org" \
-config ~/admigrate.txt \
-f -v > migrate_eng.txt
```

To simplify the migration process for multiple zones, you could put admigrate in a shell script and specify the source zone as an input variable or read it from a file with a listing of all your zones.

Inheritance and overrides

Each time you run admigrate with the same parent zone and a different source and target zone, the admigrate utility does the following:

- If a user profile from the source zone does not exist in the parent zone, the utility creates a profile for the user in the parent zone.
- If a user profile exists in the parent zone and matches the user profile from the source zone, the new child zone will inherit the user profile attributes as they are defined in the parent zone.
- If a user profile already exists in the parent zone and has attribute values that differ from those for the user from the source zone, the utility creates a user profile in the child zone with overrides for the differing attribute values. For example, if a user profile exists for oscar.romero in the parent zone, but has a different numeric identifier (UID) in the engineering zone, the UID attribute value would be different in the engineering child zones. The other attributes would be inherited from the parent zone.
- Copies the groups, rights, role definitions, role assignments, and NIS maps from the source zone to the target child zone.

The admigrate utility does not copy delegated permissions from the existing classic zones to the new child zones. In addition, delegated permissions are *not* automatically inherited from parent zones to the child zones. After migrating classic zones, you must explicitly delegate administrative permissions on a zone-by-zone basis.

Roles and rights for migrated users

The admigrate utility adds the following role definitions for migrated users:

- **login_at_roles** assigns the UNIX system rights **Password login...** and **Nonpassword login**. It does not assign **Login with non-Restricted Shell** because the user may be assigned to a restricted shell.
- **login_all_apps** assigns the login-all PAM right, which grants access to all PAM applications. It does not assign any UNIX system rights.

By default, all users are added to the **login_all_apps** role so that if they are granted login rights, they have access to all PAM applications, which is the default for users in classic zones. If PAM access rights are restricted by another role assignment, the restricted role assignment will override the rights granted by login_all_apps.

Access uses the following role-assignment rules when migrating roles and rights from a classic zone to a hierarchical zone:

User assigned to role	Enabled	Assign to the following roles: **login_at_roles**, which grants **Password login and Nonpassword login** UNIX system rights. **login_all_apps**, which grants access to all PAM applications. Corresponding user-created roles, which are migrated.
User assigned to role	Disabled	Assign to corresponding user-created roles, which are migrated. No login roles are assigned because the user is disabled in the classic zone.
User not assigned to role	Enabled	Assign to the default **UNIX Login** role, which grants all UNIX system login rights and access to all PAM applications.
User not assigned to role	Disabled	Assign to the default **listed** role, which makes the user visible in the zone but does not assign any UNIX system rights or PAM access rights.

In classic zones, users who are added to a zone are enabled for login access by default. As an administrator, you can leave a user profile defined in a zone but disable login access.

All the roles and rights you defined in the source zone, as well as any role assignments to user-created roles, are added, as-is, to the child zone each time you run admigrate. For example, if you defined a privileged mount command in 20 classic zones, admigrate will copy that mount command to 20 new hierarchical zones. Therefore, after migration you should analyze your role definitions and access right definitions to see if some of them can be moved up to the parent zone to take advantage of inheritance.

Assigning the audit level when migrating

In hierarchical zones, role definition can be assigned an auditing level. This setting is not applicable in classic zones.During migration from classic zones to hierarchical zone, the default "Audit if possible" auditing level, is assigned to all migrated role definitions. After you have migrated, you can change the auditing level in any role definition. For more information about changing the auditing level for a role definition, see Changing the audit level for role definitions.

Moving joined computers to hierarchical zones

After you have migrated data from classic zones to new hierarchical zones, you can move the computers to the new zones using the adchzone command-line program.

When you use adchzone to change the zone for a computer, the command copies the UNIX profile from the old zone to the new zone, deletes computer profile from the old zone, then stops and restarts adclient to flush the cache and update the zone information. The advantage of this approach over leaving the old zone (adleave) and then joining (adjoin) the new zone is that it is very quick and preserves all the join information without you having to specify join options.

For example, run a command similar to the following to move a computer joined to the classic finance zone to the new child finance zone:

```
/usr/share/centrifydc/adedit/adchzone \
-z "cn=finance,cn=global,cn=zones,ou=unix,dc=acme,dc=com"
-u finance-adm
```

You will be prompted to supply a password for the specified user.

After changing the zone, you can open Access Manager to see the computer in the Computers node of the new zone, or you can run adinfo on the computer to verify the new zone information.

What to do after the migration

The admigrate utility migrates most zone information automatically. After using the utility, however, you might want to perform the following tasks to complete the migration:

- Delete unnecessary copies of right and role definitions.

  You should analyze the right and role definitions to see how many of them have been copied into multiple zones. Rights and roles that are defined in the parent zone are available for use in all child zones. By moving role and right definitions to the parent zone you simplify your zone structure making it easier to understand the rights and roles that are available for your organization.

- Review provisioning rules.

  In many cases, hierarchical zones simplify automated provisioning by enabling you to define a baseline profile in a parent zone and only override specific attributes when necessary. If you are using automated provisioning, you should check whether you are defining the provisioning rules in parent zones or in child zones.

- Delegate permissions on a zone-by-zone basis.

  Use the Zone Delegation Wizard to delegate administrative tasks and the assign the corresponding permissions to the appropriate users and groups in your new child zones.

Managing Account Profiles and Identity Attributes

This chapter describes how to create user and group profiles that grant access to Centrify managed Linux and UNIX computers and how to manage identity attributes using Access Manager. No matter what type of zone your environment uses—classic, hierarchical, or auto zone—you can use Access Manager to create and maintain profiles for Active Directory users and groups.

If your environment uses hierarchical zones, you can also use Access Manager to create and maintain profiles for local Linux and UNIX users and groups.

For Active Directory users and groups, you can also perform the tasks described here using ADEdit or Windows PowerShell commands and scripts or other tools, such as Active Directory Users and Computers.

For local users and groups, you can also perform the tasks described here using adedit and other command line utilities. See Using Centrify commands for administrative tasks for details about using commands to configure local users and groups.

For additional information about planning the migration of an existing user population, see the *Planning and Deployment Guide.*

Creating Group Profiles

You can create group profiles for Active Directory groups and—in hierarchical zone environments—local groups. A group profile consists of two attributes and a list of group members. The attributes that must be defined for the group profile to be complete are the following:

- A unique numeric identifier (GID).
- A group name.

A group must have a complete profile with all of these attributes defined to be recognized as a valid group in a zone or on a specific computer. These are the same attributes you define locally for Linux and UNIX groups in the /etc/group file.

For details about creating profiles for Active Directory groups, see Creating group profiles for Active Directory groups. For details about creating profiles for local Linux and UNIX groups, see Creating, modifying, and deleting group profiles for local groups.

Creating group profiles for Active Directory groups

You can create a group profile for any domain local, global, or universal security groups you have defined in the Active Directory forest. Associating a group profile with an Active Directory group also enables you to take advantage of any nested group membership you have defined and any group policies you have applied to a domain or organizational unit.

Although associating a group profile with an Active Directory group can be convenient, there is no predetermined requirement to create group profiles for Active Directory groups. Creating a group profile does not create profiles for any members of the group. User accounts must be explicitly given their own profiles.

Note: You can automate the provisioning of account profiles through the use of Active Directory groups. For information about configuring your environment for automated provisioning, see the *Planning and Deployment Guide*.

What to do before creating a new Active Directory group profile

Before you can create Active Directory group profiles, you must have created one or more Active Directory security groups, installed Access Manager, and run the Setup Wizard. You should also identify the specific Active Directory groups for which a group profile is required. In most organizations, only a limited number of Active Directory groups require a zone profile. There are no other prerequisites for performing this task.

Rights required for this task

You must have permission to add groups to a zone. Zone administrators can grant this permission through the Zone Delegation Wizard. If the Active Directory administrator manually sets the permissions, your user account must be a domain user with the following permissions to create group profiles in a zone:

Parent container object for the group profile within the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectClass
Group account object in Active Directory For example: domain/Users/group_name	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read groupType Read objectCategory Read objectClass Read objectGUID Read objectSid
Parent container object for the individual zone For example, if you are adding a group to the Finance zone: domain/UNIX/Zones/Finance	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID Write Description

Who should perform this task

A Windows domain administrator performs this task, depending on your organization's policies. In most organizations, this task is delegated to a specific user or group with administrative authority in the selected zone.

How often you should perform this task

In most cases, you only create new group profiles infrequently to address changes to your organization.

Steps for completing this task

If you choose to create group profiles for Active Directory groups, you can use Access Manager, Active Directory Users and Computers, the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API.

The following instructions illustrate how to create a new group profile using Access Manager. Examples of scripts that use ADEdit, Windows PowerShell, or the Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a group profile for an Active Directory group using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the Active Directory group.
3. Expand UNIX Data and select Groups, right-click, then click **Create UNIX Group**.
4. Type a search string to locate the Active Directory group for which you want to create a profile, then click **Find Now**.

   For example, type "fin" to display the Finance Users and Finance Admins groups.
5. Select one or more groups in the results, then click **OK**.
6. Review the default zone profile settings for the group and make changes if needed, then click **OK**.

   You can deselect an attribute to change the default value or to create a partial group profile in the current zone. You can complete the profile by providing a value for an attribute in a child zone of the current zone. For example, if you use the same group name but different numeric identifiers on two set of computers, you can inherit the group name from a parent zone and set the different numeric identifiers in the child zones.

   In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the group's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the group's zone profile.

   If you selected more than one group, review the profile settings for the each group and modify the default settings, if necessary, then click **OK**.

   If you are adding groups with similar names, you might want to modify the default group name to distinguish the groups. For example, if you are adding both the Finance Admins and Finance Users groups to the same zone, you can change the default group name to finadmin and finuser to make it easier to tell the groups apart. Keep in mind that in some operating environments group names cannot be more than 8 characters and special characters might not be supported.

Creating, modifying, and deleting group profiles for local groups

When you create a local group profile in Access Manager, it is saved in /etc/group on each computer in each zone where the profile is defined. You can create local profiles at the zone level (for example, under **Zones** › *Zonename* › **UNIX Data**) and at the computer level (for example, under **Zones** › *Zonename* › **Computers** › *Computername* › **UNIX Data**). Local group profiles that you create at the zone level are available for local and Active Directory users in the zone and child zones to join.

What to do before creating a new local group profile

You should perform the following tasks before creating local group profiles:

- Ensure that local account management is enabled and configured through configuration parameters or group policies. See Enabling and configuring local account management for more information.
- It is suggested that you review the existing group names in etc/group on the computers where the local group profile will be implemented so that you do not attempt to create a group profile with a name that is already used. Access Manager performs a name validation check against etc/group in the current zone when you create a new local group. If the group name already exists in etc/group somewhere in the current zone, you are prompted to provide a different name for the group that you are creating.

Rights required for this task

The rights required to create local group profiles are the same as the rights required to create Active Directory group profiles. See Rights required for this

task for details about those rights.

Using partial profiles and child zones to fine tune group attributes

Access Manager allows you to create a partial profile by leaving any of the attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could define the Members attribute in a parent zone, and then override the parent zone attribute settings by defining the Members attribute differently in different child zones.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to create the group profile.

Groups can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone. If a group profile is still partial at the computer level, the profile is ignored by the agent, and it is not added to /etc/group on the local computer. Group profiles must contain the attributes listed in Creating group profiles to be complete.

Specifying profile states

The *profile state* lets you control whether a local group account is in place in etc/group and is enabled for use locally. When you create a local group account, you specify the initial profile state. You can change the profile state afterwards to control availability of the local group account. A local group account can have one of the following states:

- **Enable**: If the local group profile is complete, it will be installed or updated in /etc/group at the next local account refresh interval.
- **Remove from /etc/group**: The group profile will be removed from etc/group at the next local account refresh interval.

You can also choose not to define the profile state by deselecting the **State** check box in the Set Local Group Profile dialog. Deselecting the **State** check box results in one of the following scenarios:

- If a local group profile with the same name exists in the parent zone, the state from the parent group profile is inherited.
- If the parent zone does not contain a group profile with the same name, or if a parent group profile exists but does not define the state, the group profile that you are currently defining is considered incomplete.

Roles and local group account visibility

You use role assignments to control whether local users are visible in a zone. A predefined role definition, local listed, is available for use with local user and local group profiles. As with the listed predefined role, the local listed role does not grant any system rights, PAM rights, or command rights. It is a specialized role that can be used when a local user or local group profile must exist for computers in a zone, but no local user or local group access should be granted.

You can optionally define other roles in the zone to grant visibility to local users and local groups.

By default, all local groups having a complete profile are visible in a zone. You do not have to assign a role to a local group to make the local group visible. However, it is often useful to assign a role (such as local listed) to a local group so that all local users in the local group inherit the role assignment, and are visible in the zone.

See Creating, modifying, and deleting user profiles for local users for more information about how roles are used to control visibility of local user accounts.

How often Access Manager and local group accounts are synchronized

The /etc/group file on local computers is updated periodically based on the information that you define for local group profiles in Access Manager. The /etc/group update interval is controlled by the following group policy and configuration parameter:

- **Group Policy: Set refresh interval for access control cache**, located in Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings.
- **Configuration parameter:** adclient.refresh.interval.dz, located in the /etc/centrifydc/centrifydc.conf configuration file.

The same group policy and parameter control how often the authorization store cache is updated. Local account information is updated immediately after authorization store information is refreshed in the authorization cache.

For more information, see the *Group Policy Guide*, the *Configuration and Tuning Reference Guide*, and Enabling and configuring local account management.

Steps for completing this task

To create a group profile for a local group using Access Manager

1. Open Access Manager.
2. Expand Zones and any parent zones, child zones, or computers required to select the zone or computer to which you want to add the local group.
3. Expand UNIX Data and select Local Groups.

   You can create a new local group in these ways:

   - **By dragging and dropping an existing local group from another location.** Expand zones or computers to the location of the original local group, and drag it to the location of the new local group. The local group is moved to the new location. To copy (instead of move) the original group, press <Ctrl> while you drag the group.
   - **By cutting or copying an existing local group from another location, and then pasting it into the current location.** Expand zones or computers to the zone where the original local group exists, right-click a local group and select **Cut** or **Copy**, return to the zone where you are creating the new local group, right-click, and select **Paste**.
   - **By creating an entirely new local group.** Perform Step 4 through Step 8 of this procedure.
4. In Local Groups, right-click, then click **Create UNIX Group**.
5. Type a name for the new local group and click **OK**.
6. In the Set UNIX Group Profile dialog, select or deselect check boxes to specify which attributes to set. You must specify at least one attribute to be able to save the profile.

   - **GID:** Type a numeric group ID of your choice.
   - **Members:** Click **Add** to launch the Add Members dialog. In a comma-separated list, type the UNIX names of the users who will be in the group.

     Access Manager does not check the validity of the user names that you provide. You should ensure that all of the names that you provide are UNIX names that currently exist.

     Note that the group profile is considered complete even if this attribute has an empty value.
   - **State:** Specify whether the group account is added to, and enabled in, etc/group. Possible values are:

     Enable: The group profile will be installed or updated in /etc/group at the next local account refresh interval.

     Remove from /etc/group: The group profile will be removed from etc/group at the next local account refresh interval.

     Note: To modify permissions for a local group, you must first
     create and save the local group as described in this procedure, and then
     modify permissions as described in Step 4 in the section(To modify group
     profile attributes and permissions for a local group](https://docs.centrify.com/Content/auth-admin-unix/ProfilesGroupLocal.htm#modify_permissions_local_group).

     For the profile to be complete, it must contain settings for group name (specified in Step 5 of the procedure To create a group profile for a local group using Access Manager), GID, and state. You can save the profile now even if it is partial, although it will not be implemented in /etc/group until you update it in the current zone, or with settings in child zones, so that it is complete, and you set the state to **Enable**. For example, if you use the same group name but different numeric identifiers on two set of computers, you can inherit the group name from a parent zone and set the different numeric identifiers in the child zones.

     In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the local group's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the group's zone profile.
7. Review your local group profile settings and click **OK**.

   If the profile is complete, it is added to /etc/group at the next local account refresh interval.
8. To optionally assign the local listed role to the local group, so that all local users in the local group are visible in the zone:

   1. At the level where you created the local group, right-click **Role Assignments**, and then select **Assign Role**.

2. In the Select Role dialog, select **local listed** and click **OK**.
3. In the Assign Role dialog, ensure that **Accounts below** is selected, and click **Add Local Account**.
4. In the **Add Local Account** dialog, select **Local UNIX Group** in the **Type** field, type the local group name in the **Account** field, and click **OK**.
5. In the Assign Role dialog **Accounts below** area, highlight the local group account and click **OK**. The local group is now listed as an assignee of the local listed role.

To modify group profile attributes and permissions for a local group:

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to modify.
2. In the Local Groups details pane, right-click the local group to modify and select **Zone Profile**.

   The Properties dialog for the profile is displayed.
3. Modify attribute selections and settings as described in Step 6 in the procedure To create a group profile for a local group using Access Manager. Keep in mind the following considerations when you change attributes.

   If there is no parent profile for the same local group name:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to define a partial profile.

     If a parent profile for the same local group name already exists in a parent zone:
   - You can edit profile fields to customize the value.
   - You can deselect profile fields to inherit attribute values from the parent profile.
4. To optionally modify group permissions (such as read, write, create or delete child object, and so on), click **Permissions**. Refer to the "Active Directory permissions required for administrative tasks" chapter in the *Planning and Deployment Guide* for details about using the Permissions dialog to modify zone-level user and group permissions.
5. Review your changes to the local group profile and click **OK**.

   Your changes are applied to the local group profile in /etc/group at the next local account refresh interval.

To delete a group profile for a local group from a zone or computer:

Note: This procedure does not remove a local group profile from /etc/group. To remove a local group profile from /etc/group, perform the procedure described in [To remove a group profile for a local group from /etc/group](https://docs.centrify.com/Content/auth-admin-unix/ProfilesGroupLocal.htm#remove_group_profile).

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to delete.
2. In the Local Groups details pane, right-click the local group to modify and select **Delete**.
3. At the warning prompt, select **Yes**.

   The local group is deleted from Access Manager. The group profile still exists in /etc/group, but it is ignored.

To remove a group profile for a local group from /etc/group

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to remove from /etc/group.
2. Perform one of the following procedures:

   - Right-click a local group, select **Change Profile State**, then select **Remove from /etc/group**.
   - Right-click a local group, select **Zone Profile**, change the value of the **State** field to **Remove from /etc/group**, and click **OK.**

     At the next local account refresh interval, the local group's profile is removed from /etc/group.

Delegating control of local group management tasks

You can use the Zone Delegation Wizard and Computer Delegation Wizard as described in the *Planning and Deployment Guide* to delegate control of local group management tasks.

Delinea

Migrating Local Group Profiles to Active Directory

In most cases, you get more operational benefits by using Active Directory groups to manage UNIX and Linux user accounts than you would get from migrating your local group profiles into Active Directory. For example, by using Active Directory groups to manage both Windows and UNIX users, you can use your existing provisioning and access control policies across multiple platforms and automate the provisioning and de-provisioning of accounts and access rules.

In some cases, however, you might find it useful to migrate some or all of your existing local groups to Active Directory. If you want to move local group profiles into Active Directory, you have the option to import local groups on a zone-by-zone basis. As part of the import process, you can choose to how each local group should be handled. For example, you can:

- Create a new Active Directory group for each imported group.
- Extend an existing Active Directory group to include an imported group.
- Merge an imported group into an existing UNIX group profile.

Making Group Membership a Requirement

On most Linux and UNIX computers, users can only be members of a limited number of groups at once. Because of this limitation, it is useful to be able to change a user's effective group membership to add and remove groups when necessary. You can use the adsetgroups command to dynamically manage the set of Active Directory groups that are available to a user account. You also have the option to specify that membership in a specific group is required in a zone. If you specify that a group is required, users who are members of the group cannot remove the required group profile from their currently active set of groups.

To make membership in a specific group profile required:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name for which you want to add a required group.
3. Expand Groups, then select the group name you want to make required.
4. Right-click, then select **Zone Profile** to display the Centrify UNIX Profile for the group.
5. Select the **Users are required to be members of this group** option.
6. Click **Permissions** to set specific permissions for this group, if needed, then click **OK**.

For more information about using the adsetgroups command, see the adsetgroups man page.

Creating User Profiles

You can create user profiles for Active Directory users and—in hierarchical zone environments—local users. A user profile consists of the attributes required by the name service switch (NSS) facility on Linux and UNIX computers. User attributes that must be defined for the user profile to be complete are the following:

- A user name (the UNIX login name).
- A unique numeric user identifier (UID).
- The user's primary group profile numeric identifier (GID).
- The default home directory for the user.
- The default login shell for the user.
- General information about the user account (GECOS). (This attribute is required for Active Directory user profiles, but not for local user profiles.)

A user must have a complete profile with all of these attributes defined to be recognized as a valid user in a zone or on a specific computer. You can optionally define other attributes that are not required for the user profile to be complete.

These are the same attributes you define locally for Linux and UNIX users in the /etc/passwd file.

For details about creating profiles for Active Directory users, see Creating user profiles for Active Directory users. For details about creating profiles for local Linux and UNIX groups, see Creating, modifying, and deleting user profiles for local users.

Creating user profiles for Active Directory users

You can create a user profile for any domain user you have defined in the Active Directory forest by adding the user to a zone, or by adding the user to a specific computer in a zone. Associating a user profile with an Active Directory user determines how the Active Directory user is identified on Linux and UNIX computers.

Note: You can automate the provisioning of user profiles through the use of Active Directory groups. For information about configuring your environment for automated provisioning, see the *Planning and Deployment Guide*.

What to do before creating a new Active Directory user profile

Before you can create Active Directory user profiles, you must have created one or more Active Directory users, installed Access Manager, and run the Setup Wizard. You should also identify the computers where Active Directory users might require different profile attributes. For example, you might have some Active Directory users that require the default home directory attribute to be set the to /home for access to most computers, but require the attribute to be set to /Users when they log on to Mac OS X computers.

In most organizations, Active Directory users have one “dominant” profile with consistent attributes across multiple computers, but require “override” settings to some profile attributes on specific computers or groups of computers. Therefore, most user profiles are only added to parent zones and inherited in child zones.

Rights required for this task

You must have permission to add users to a zone. Zone administrators can grant this permission through the Zone Delegation Wizard. If the Active Directory administrator manually sets the permissions, your user account must be a domain user with the following permissions to create user profiles in a zone:

Parent container object for the user profile	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create serviceConnectionPoint Objects This permission is required for both standard zones and RFC 2307compliant zones. For standard zones, you need to apply additional permissions. Click the **Properties** tab and select **serviceConnectionPoint objects** from the object list, then select **Allow** to apply the following properties to this object: Read Name Read name Read displayName
User account object in Active Directory For example: domain/Users/user_name	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectCategory Read objectClass Read objectGUID Read objectSid Read userAccountControl

Parent container object for the individual zone For example, if you are adding a user to the Finance zone: domain/UNIX/Zones/Finance	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read objectGUID Write Description

Who should perform this task

A Windows domain administrator performs this task, depending on your organization's policies. In most organizations, this task is delegated to a specific user or group with administrative authority in the selected zone.

How often you should perform this task

In most cases, you create and remove user profiles frequently to address changes to your user population.

Steps for completing this task

The following instructions illustrate one way to create a new user profile using Access Manager. You can also add a user profile and assign a role to an Active Directory user with the Add User wizard. Examples of scripts that use ADEdit, Windows PowerShell, or the Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a user profile for an Active Directory user using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the Active Directory group.

   In most cases, you should add user profiles to a parent zone.
3. Expand UNIX Data and select **Users**, right-click, then click **Add User to Zone**.
4. Type a search string to locate the user account, then click **Find Now**.

   For example, type "qa" to display the qa-lab, qa-hk and qaVenice1x users.
5. Select one or more users in the results, then click **OK**.
6. Review the default zone profile settings for the user and make any changes if needed, then click **OK**.

   You can deselect an attribute to change the default value or to create a partial user profile in the current zone. You can then complete the profile by providing a value for an attribute in a child zone of the current zone. For example, if you use the same login name but different numeric identifiers on two set of computers, you can inherit the login name from a parent zone and set the different numeric identifiers in the child zones.

   In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the user's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the user's zone profile.

   If you selected more than one user, review the profile settings for the each user and modify the default settings, if necessary, then click **OK**.

Changing the default profile attributes

When you add Active Directory users to a zone, Access Manager displays a default new user profile. You can accept or change the default values for any of the profile attributes, as needed. The default attribute values are automatically generated based on a few simple rules and, in most cases, you can accept them as-is. The following table describes how the default values are populated.

Login name	The Active Directory user logon name associated with the Active Directory account.

UID	A unique number automatically generated by an algorithm based on the security identifier (SID) for the Active Directory user.
Primary group	A unique numeric identifier that represents a private primary group and is the same as the user's default UID. Private groups are not stored or managed in Active Directory.
GECOS	A runtime variable that resolves to the Active Directory displayName attribute associated with the Active Directory account.
Home directory	A runtime variable that specifies the default home directory when resolved locally on a computer.
Shell	A runtime variable that specifies the default login shell when resolved locally on a computer. To set the user's shell to the default shell defined for this computer in this zone.

Defining partial UNIX profiles

Access Manager allows you to create a partial profile by leaving any of the attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could leave the Shell attribute blank in a parent zone, define it as /bin/bash in a child zone, but override it with /usr/bin/ksh in a grandchild zone that only contains AIX computers. You could also leave the Home directory attribute blank in a parent zone, then set it to /home in one child zone and to /Users on an individual Mac OS X computer that joins the child zone.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to add the user profile. Users must have a complete profile in a zone for any role assignments to be effective. Keep in mind, however, that users can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone to allow role assignments in the child zone.

Defining valid login names

User profile login names can consist of letters, numbers, hyphens, underscores, periods and dashes. Some operating environments may have additional restrictions. For example, some operating environments do not support user names that are longer than 8 characters or require that the first character of the user name be alphabetic. Because UNIX user names typically use only lowercase characters, the default user profile name displayed follows this convention. If you modify the default profile name and include uppercase characters, keep in mind that the proper case must be used when entering the user name. For compatibility with Samba, the dollar sign ($) can also be used at the end of the user name. In general, other special characters, such as ! and &, are not supported.

If the Windows logon name includes unsupported special characters, Access Manager replaces them with underscores for the UNIX login name. For example, Access Manager converts a Windows logon name with special characters, such as qa:user2 into a valid UNIX login name of qa_user2.

Identifying a primary group

In most UNIX environments, a user's primary group identifier (GID) is a “private” group that exists solely for that user. The user is not included as a “member” of the private primary group. You can follow this convention by using a UNIX-only “private” group that is not linked to an Active Directory group, which is the default when you create a new user profile.

If you keep the default private primary group, the primary group identifier (GID) setting in the user profile does not affect the user's actual Active Directory group membership in any way, and there's no need to manage primary groups for UNIX users through Active Directory.

In some cases, however, you might want to assign an Active Directory group that has a corresponding group profile as a user's primary group. If you specify an Active Directory group as a user's primary group, keep in mind that you must manage the membership of that group using Active Directory Users and Computers and that if you identify a group with a large number of members—such as Domain Users—it is likely to affect performance.

For more information about defining primary groups for users, see the *Planning and Deployment Guide*.

Creating, modifying, and deleting user profiles for local users

When you create a local user profile in Access Manager, it is saved in /etc/passwd on each computer in each zone where the profile is defined. You can create local profiles at the zone level (for example, under **Zones** › *Zonename* › **UNIX Data**) and at the computer level (for example, under **Zones** › *Zonename* ›

Computers > *Computername* > **UNIX Data**).

After you create local user profiles, you perform a separate set of tasks to create and manage local user passwords. For detailed information about local user passwords, see Creating and managing local user passwords.

What to do before creating a new local user profile

You should perform the following tasks before creating local user profiles:

- Ensure that local account management is enabled and configured through configuration parameters or group policies. See Enabling and configuring local account management for more information.
- It is suggested that you review the existing user names in etc/passwd on the computers where the local user profile will be implemented so that you do not attempt to create a user profile with a name that is already used. Access Manager performs a name validation check against etc/passwd in the current zone when you create a new local user. If the user name already exists in etc/passwd somewhere in the current zone, you are prompted to provide a different name for the user that you are creating.

Rights required for this task

The rights required to create local user profiles are the same as the rights required to create Active Directory user profiles. See Rights required for this task for details about those rights.

Using partial profiles and child zones to fine tune user attributes

Access Manager allows you to create a partial profile by leaving some user attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could leave the Shell attribute blank in a parent zone, define it as /bin/bash in a child zone, but override it with /usr/bin/ksh in a grandchild zone that only contains AIX computers.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to create the user profile.

Users can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone. If a user profile is still partial at the computer level, the profile is ignored by the agent, and it is not added to /etc/passwd on the local computer. User profiles must contain the attributes listed in Creating user profiles to be complete.

Specifying profile states

The *profile state* lets you control whether a local user account is in place in etc/passwd and is enabled for use locally. When you create a local user account, you specify the initial profile state. You can change the profile state afterwards to control availability of the local user account. A local user account can have one of the following states:

- **Enable**: If the user profile is complete, it will be installed or updated in /etc/passwd at the next local account refresh interval. The user can log into the local computer, and is visible in Access Manager if a role with the visible right (such as local listed) is granted to the user. See Roles and local user account visibility for more information about how roles affect local user visibility.
- **Disable**: If the user profile is complete, it will be installed or updated in /etc/passwd at the next local account refresh interval. However, the user will not be able to log into the local computer. This state results in what is typically called a "locked account." UNIX and Linux service accounts and system accounts are typically set up as locked accounts.
- **Remove from /etc/passwd**: The user profile will be removed from etc/passwd at the next local account refresh interval.

You can also choose not to define the profile state by deselecting the **State** check box in the Set Local User Profile dialog. Deselecting the **State** check box results in one of the following scenarios:

- If a local user profile with the same name exists in the parent zone, the state from the parent user profile is inherited.
- If the parent zone does not contain a user profile with the same name, or if a parent user profile exists but does not define the state, the user profile that you are currently defining is considered incomplete.

Roles and local user account visibility

You use role assignments to control whether local users are visible in a zone. A predefined role definition, local listed, is available for use with local user and local group profiles. As with the listed predefined role, the local listed role does not grant any system rights, PAM rights, or command rights. It is a specialized role that can be used when a local user profile must exist for computers in a zone, but no local user access should be granted.

You can optionally define other roles in the zone to grant visibility to local users.

As with role assignments for Active Directory users, local user role assignments can be made at the zone level, computer level, or computer role level. Use the following guidelines to establish where local users are visible in Access Manager:

- To make a local user visible to all computers in a zone, assign the local listed role to the local user account (or to all local UNIX accounts) in the zone (for example, assign local listed to users located in **Zones** › *Zonename* › **UNIX Data** › **Local Users**).
- To make a local user visible only to a specific computer, assign the local listed role to the local user account (or to all local UNIX accounts) located in the computer zone (for example, assign local listed to users located in **Zones** › *Zonename* › **Computers** › *Computername* › **UNIX Data** › **Local Users**).
- To make a local user visible only to a group of computers, create a computer role and assign the local listed role to the local user account (or to all local UNIX accounts) in the computer role.

How often Access Manager and local user accounts are synchronized

The /etc/passwd file on local computers is updated periodically based on the information that you define for local user profiles in Access Manager. The /etc/passwd update interval is controlled by the following group policy and configuration parameter:

- **Group Policy: Set refresh interval for access control cache**, located in Computer Configuration › Centrify Settings › DirectControl Settings › Network and Cache Settings.
- **Configuration parameter:** adclient.refresh.interval.dz, located in the /etc/centrifydc/centrifydc.conf configuration file.

These are the same group policy and parameter that control how often the authorization store cache is updated. Local account information is updated immediately after authorization store information is refreshed in the authorization cache.

For more information, see Enabling and configuring local account management of this guide. For additional group policy and configuration parameter information, see the *Group Policy Guide*, and the *Configuration and Tuning Reference Guide*.

Steps for completing this task

To create a user profile for a local user using Access Manager, Method 1

Note: This method begins from the Local Users node, and allows you to assign just one role, local listed, to the local user. To use the Add User to Zone wizard, which lets you assign other roles to the local user, see [To create a user profile for a local user using Access Manager, Method 2](https://docs.centrify.com/Content/auth-admin-unix/ProfilesCreatingLocal.htm#user_profile_method_2).

1. Open Access Manager.
2. Expand Zones and any parent zones, child zones, or computers required to select the zone or computer to which you want to add the local user.
3. Expand UNIX Data and select Local Users.

   You can create a new local user in these ways:

   - **By dragging and dropping an existing local user from another location.** Expand zones or computers to the location of the original local user, and drag it to the location of the new local user. The local user is moved to the new location, and no longer exists in the original location. To copy the original user to the new location and also retain it in the original location, press <Ctrl> while you drag the user.
   - **By cutting or copying an existing local user from another location, and then pasting it into the current location.** Expand zones or computers to the zone where the original local user exists, right-click a local user and select **Cut** or **Copy**, return to the zone where you are creating the new local user, right-click, and select **Paste**.
   - **By creating an entirely new local user.** Perform Step 4 through Step 8 of this procedure.
4. In Local Users, right-click, then click **Add User to Zone**.

5. Type a name for the new local user and click **OK**.
6. In the Set UNIX User Profile dialog, select or deselect check boxes to specify which attributes to set. You must specify at least one attribute to be able to save the profile.

   If a parent profile for the same local user name already exists in a parent zone, some attribute fields will be filled in already with inherited values. You can edit profile fields to customize inherited values, and you can deselect other profile fields to inherit attribute values from the parent profile.

   - **UID:** Type a numeric user ID of your choice.
   - **Primary group:** From the drop-down list, select an existing group, or select **<Not defined>** to leave the PGID attribute undefined, or select **<...>** to see additional group choices or to create a new local group.

     To create a new local group after clicking **<...>**, click **Add** in the Select a Group dialog, and follow the procedure for creating a new local group starting with Step 5 in the section To create a group profile for a local group using Access Manager.
   - **GECOS:** Optionally type general information of your choice about the local user account. This attribute is not required for the profile to be complete.
   - **Home directory**: Type the default local computer home directory for the local user.
   - **Shell**: Select the default shell for the local user. Choices are /bin/bash, /bin/csh, /bin/ksh, /bin/sh, /bin/tcsh, %.
   - **State:** Specify whether the local user account is added and enabled in /etc/passwd. Choices are as follows.
     - **Enable**: If the user profile is complete, it will be installed or updated in /etc/passwd at the next local account refresh interval. The user will be able to log into the local computer, and the user is visible in Access Manager.
     - **Disable**: If the user profile is complete, it will be installed or updated in /etc/passwd at the next local account refresh interval. However, the password field in /etc/passwd will be set to !!, and the user will not be able to log into the local computer. This state results in what is typically called a "locked account." The user is still visible in the zone as long as the local listed role is assigned to the user.
     - **Remove from /etc/passwd**: The user profile will be removed from etc/passwd at the next local account refresh interval.

     For the profile to be complete, it must contain the attributes listed in Creating user profiles. You can save the profile even if it is partial, although it will not be implemented in /etc/passwd until you update it in the current zone, or with settings in child zones, so that it is complete, and you set the state to enabled. For example, if you use the same user name but different numeric identifiers on two set of computers, you can inherit the user name from a parent zone and set the different numeric identifiers in the child zones.

     In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the local user's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the user's zone profile.

     Note: To modify permissions for a local user, you must first create and save the local user as described in this procedure, and then modify permissions as described in [To modify user profile attributes and permissions for a local user](https://docs.centrify.com/Content/auth-admin-unix/ProfilesCreatingLocal.htm#modify_profile_attributes).
7. By default, new local users are assigned the local listed role so that local users are visible in Access Manager. This assignment is specified in the **Assign local listed role to make this user visible** check box. To keep this default assignment, ensure that the check box remains selected.

   To give the local user a different role assignment, deselect the check box. If you deselect the check box, you will need to manually assign a role with visible rights to the local user after completing this procedure.
8. Review your attribute selections and settings, and click **OK**. If the user profile is complete, it is added to /etc/passwd at the next local account refresh interval.

To create a user profile for a local user using Access Manager, Method 2

Note: This method describes how to create a local user profile using the Add User to Zone wizard, which lets you assign roles other than just local listed to the local user.

1. Open Access Manager.
2. Expand Zones and any parent zones, child zones, or computers required to select the zone to which you want to add the local user.
3. Right-click the zone, and select **Add User**.

The Add User to Zone wizard launches.

4. In the Select User Type dialog, select **Local UNIX user**, and click **Next**.
5. In the Specify Local UNIX User dialog, type a name for the local user, and click **Next**.
6. In the Add User to Zone dialog, select the **Define user UNIX profile** and **Assign roles** check boxes. Click **Next**.
7. Fill in the local users profile attribute settings in the Define User UNIX Profile dialog as described in Step 6 in the section To create a user profile for a local user using Access Manager, Method 1, and click **Next**.
8. In the Assign Roles dialog, the local listed role is included by default. To optionally add different roles as choices, click **Add** and select one or more roles to add to the list.
9. In The Assign Roles dialog, select one or more roles, and click **Next**.
10. In the Confirm Your Selections dialog, review your choices and click **Next**.
11. In the final wizard screen, click **Finish**.
12. Confirm that the new local user was created by expanding UNIX Data in the zone and clicking **Local Users**. The new local user should be listed in the user details pane.

To modify user profile attributes and permissions for a local user:

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to modify.
2. In the Local User details pane, right-click the local user to modify and select **Zone Profile**.

   The Properties dialog for the profile is displayed.
3. Modify attribute selections and settings as described in Step 6 in the section To create a user profile for a local user using Access Manager, Method 1. Keep in mind the following considerations when you change attributes.

   If there is no parent profile for the same local user name:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to define a partial profile.

     If a parent profile for the same local user name already exists in a parent zone:
   - You can edit profile fields to customize the value.
   - You can deselect profile fields to inherit attribute values from the parent profile.
4. To optionally modify user permissions (such as read, write, create or delete child object, and so on), click **Permissions**. Refer to the "Active Directory permissions required for administrative tasks" chapter in the *Planning and Deployment Guide* for details about using the Permissions dialog to modify zone-level user and group permissions.
5. Review your changes to the local user profile and click **OK**.

   Your changes are applied to the local user profile in /etc/passwd at the next local account refresh interval.

To disable a user profile for a local user:

Note: This procedure does not remove a local user profile from /etc/passwd. To remove a local user profile from /etc/passwd, perform the procedure described in To remove a user profile for a local user from /etc/passwd.

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to disable.
2. In the Local Users details pane, right-click the local user and select **Change Profile State**.
3. Select **Disable**.

The local user remains visible in Access Manager. At the next local account refresh interval, the local user's profile in /etc/passwd is modified so that the password field contains !!, and the user cannot log into the local computer.

To delete a local user from a zone

Note: This procedure does not remove a local user profile from /etc/passwd. To remove a local user profile from /etc/passwd, perform the procedure described in "To remove a user profile for a local user from /etc/passwd before you delete the local user from the zone.

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to delete from the zone.
2. In the Local Users details pane, right-click the local user and select **Delete**.
3. In the confirmation dialog, select **Yes** to delete the user from the zone. To prevent the confirmation dialog from displaying in the future, select **Do not warn me again**.
4. In the next confirmation dialog, select **Yes**.

   The user is removed from zone, and is no longer controlled through Access Manager. However, the user profile remains in /etc/passwd on local computers.

To remove a user profile for a local user from /etc/passwd

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to remove.
2. In the Local Users details pane, right-click the local user and select **Change Profile State**.
3. Perform one of the following procedures:
   - Right-click the local user, select **Change Profile State**, then select **Remove from /etc/passwd**.
   - Right-click the local user, select **Zone Profile**, change the value of the **State** field to **Remove from /etc/passwd**, and click **OK.**

At the next local account refresh interval, the local user's profile is removed from /etc/passwd.

Delegating control of local user management tasks

You can use the Zone Delegation Wizard and Computer Delegation Wizard as described in the *Planning and Deployment Guide* to delegate control of local user management tasks.

Creating and managing local user passwords

After you create local user profiles as described in the preceding sections, you still need to assign a password to each user. You can create local user passwords in one of these ways:

- By creating a shell script to execute the passwd command on each local computer, giving each local user the password that you specify in the script. The shell script can be executed manually, or by enabling adclient.local.account.notification.cli to run the script automatically when local accounts are refreshed. This is the least secure way to assign passwords to local users, because the same password is assigned to each user when the script runs. After the script runs, you must change passwords locally so that each password is unique.

  This guide does not include detailed instructions for implementing this method of creating local user passwords.

- If your environment contains a third-party password management product, you can create a shell script that executes on each local computer, giving each local user a random password. The shell script can include a section that submits the passwords to the password management product for storage and maintenance. The shell script can be executed manually, or by enabling adclient.local.account.notification.cli to run the script automatically when local accounts are refreshed.

  A sample shell script, handle_local_accts.sh, is provided in /usr/share/centrifydc/samples/localacctmgmt for you to use as a reference when you create your own shell script. Typically, the shell script that you create should perform the following tasks:

  - Assign a random password to newly provisioned local users, and to local users whose accounts were recently unlocked (that is, re-enabled after having been disabled).
  - Optionally create a home directory for each new local user.

- Provide the user account information, including the generated passwords, to a third-party password management solution.

  For syntax details about the notification CLI, execute the sample script with the -h option:

  handle_local_accts.sh -h

- If your environment does not contain a third-party password management product and you want to create and maintain unique passwords for each local user, you can use Server Suite to manage local user passwords.

  Using Server Suite to manage local user passwords involves these tasks:

  - Register for Server Suite.
  - Download the Centrify Agent for Linux software package.
  - On each UNIX and Linux computer where you will assign passwords to local users, execute the cenroll command to register the computer as a managed resource.
  - Create a shell script that executes on each local computer, giving each local user a random password. The shell script should include commands to manage generated passwords. The agent package includes a sample shell script that you can use as a reference when you create your own shell script.
  - Enable the adclient.local.account.notification.cli configuration parameter to run the shell script automatically when local accounts are refreshed.

Setting Runtime Variables in User Profiles

Access Manager maintains a set of predefined runtime variables that you can use in place of specific values in Active Directory user profiles and local user profiles. Using the variables simplifies the process of defining profile attributes. The Centrify Agent for *NIX resolves the runtime variables defined in a profile with appropriate values when a computer joins a domain and zone.

The predefined runtime variables you can use in profiles are:

%	The domain to which the computer is joined.
%	The root home directory. By default, this directory is /home on most Linux and UNIX computers. For Mac OS X computers, the default home directory is /Users. On Solaris computers, the default home directory is /export/home).
%	The host name of the joined computer.
%	The default login shell for the user. By default, the shell is /bin/bash on most Linux and UNIX computers. On Solaris and HP computers, the default shell is /bin/sh. On AIX computers, the default shell is /usr/bin/ksh.
%	The Active Directory site of the joined computer.
%	The user's UNIX login name. **Note:** This variable is supported only for Active Directory users. It is not supported for local users.
%	The zone to which the computer is joined.

You can use these predefined runtime variables or custom variables at any point in the zone hierarchy, including a parent zone, a child zone, or on individual computers. At runtime, the adclient process resolves the variables based on how the following configuration parameters are set and where the variables are defined in the zone hierarchy:

- nss.runtime.defaultvalue.var.variableName

  These parameters — one for each predefined variable — defines the default value for each parameter as shown in the table. These are the values are used if the variable is not explicitly defined in the zone or by the nss.runtime.var.variableName parameter in the configuration file. For example:

  nss.runtime.defaultvalue.var.home: /home

  nss.runtime.defaultvalue.var.shell: /bin/bash

- nss.runtime.var.variableName

  These parameters allow you to specify a specific value for any of the predefined variables in the configuration file. The value in the configuration file is essentially a computerspecific override because it applies only to the computer on which it is defined and overrides any other setting for the variable, including the default value, or a specific value in a zone Properties page. For example:

  nss.runtime.var.home: /Users

  nss.runtime.var.shell: /bin/sh

To override the default definition for any predefined variable in a zone, you can simply add a variable with the same name to the zone by using the zone Properties page or by using ADEdit. Zone variables and zone variable definitions are inherited down the profile tree, which means that a variable could have one definition at the top of the tree and a different definition at the bottom. The value that is applied depends at which level of the zone hierarchy a computer joins the domain.

To define values for predefined variables in a parent or child zone:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name in which you want to override a profile attribute.

For example, if you want to override the default login shell in the child zone that only AIX computers join, you might expand Child Zones to view and select the IBM AIX Only zone.

3. Select the zone, right-click, then click **Properties**.
4. Click the **Variables** tab, then click **Add**.
5. Type the name of the predefined variable and the custom value you want to use, then click **OK** to save the variable definition.

   For example, type shell and set the value to /usr/bin/ksh to modify the default shell definition.
6. Click **OK** to close the zone properties.

Using Active Directory attributes as variables

You can also use any Active Directory user attributes as variables by specifying the attribute name in the following format:

%

For example, if you want to populate the GECOS field of a user's zone profile with the information from the user's department attribute, you could specify the variable as follows:

%

By default, only a subset of common user object attributes can be retrieved and resolved by the adclient process. The default set of attributes you can use in a user profile are:

- mail
- department
- description
- mobile
- title
- telephoneNumber

The most common format for the GECOS field in a user profile contains the user's full name, building number, and office phone number separated by commas. Depending on the operating system and desktop manager you are using, the information from the GECOS field might also be used to display the user name when logging on. If you specify an attribute for the GECOS field that includes a comma, you might see the first part of the attribute treated as the user's full name and displayed in the login screen. For example, if you are using the department attribute in the GECOS field and the attribute is defined as "Cendura, San Francisco, Engineering, 25th floor, office 202", you might see Cendura listed as a user on the login screen.

Using other attributes in a profile

The default user attributes are recognized by adclient without requiring any modification to the managed computer or Active Directory. If you want to use any other attribute, whether it is a standard schema attribute like company or homePhone or a custom attribute that you have added to the Active Directory schema such as supervisorId, you must add an entry for the attribute to the adclient.custom.attributes.user parameter in centrifydc.conf file, then restart adclient and flush the cache.

For example, you might add the following attributes to the centrifydc.conf file:

adclient.custom.attributes.user: company supervisorId

After modifying the file, you would run the following commands to restart the agent and clear the cache:

/usr/share/centrifydc/bin/centrifydcrestart

adflush -f

For more information about defining custom attributes, see the *Configuration and Tuning Reference Guide*.

Attributes for users in a forest with a one-way trust

Keep in mind when using attribute variables that if you add users to a zone from a oneway trusted forest, the Centrify Agent will only be able to retrieve values for the userPrincipalName and samAccountName attributes. Therefore, at runtime, when the adclient process resolves variable definitions, fields that contain any other variables will be blank for a user from a one-way trusted forest.

Adding custom variables to a zone

You can also create your own variables at any point in the zone hierarchy, including a parent zone, a child zone, or on individual computers. You can add custom variable names and values in exactly the same way you define new values for the predefined runtime variables, except that you type a custom variable name and value.

Importing Local Account Profiles

Most organizations have at least some local user and group profiles that must be migrated to Active Directory. Access Manager provides an Import from UNIX wizard that enables you to import user and group profiles from local /etc/passwd and /etc/group files or from NIS servers and domains.

If you are not migrating any local account profiles, you can skip this section. However, if you have a large or complex user population to migrate, you should use the information in this section along with the *Planning and Deployment Guide* for a more complete view of the migration process and analysis requirement.

Collecting account information

Before using the Import from UNIX wizard, you should do the following to prepare:

- Identify each source of user information and analyze the information to determine your zone requirements.
- Run appropriate commands—such as getent passwd, getent group, or niscat—to export user and group information and save it in properly-formatted text files.

  Copy the text files to a location that is accessible from the Windows network. If you want to import information directly from NIS maps instead of text files, you should verify that you can access NIS servers and domains from the Windows network.
- Review the text files entries to remove account entries that don't need to be mapped to Active Directory accounts.

  You can automatically exclude system accounts with UID or GID values from 0 to 99 during the import process, but you might want to remove other accounts prior to the import. As part of the review process, determine which entries should map to existing Active Directory accounts or which entries require new Active Directory objects.

Using variables when importing UNIX users

When you import UNIX user accounts, you can use a variable in the GECOS field so that Active Directory will automatically populate that information. The variable you can use is as follows:

%

For example: In your /etc/passwd file, you have the following information for a user:

ron:x:10061:10061:%:/home/ron:/bin/bash

After you import the user with the UNIX import user wizard, the following user is in the pending import area:

UID: 10061

Login name: ron

Shell: /bin/bash

Home directory: /home/ron

Primary Group: 10061

GECOS: %

After you map this pending user to a user account in Active Directory, the % text is converted to the user's display name at runtime by adclient. When you view the user profile in Active Directory or Access Manager, you'll see the % text in the GECOS field; when you query the user from a UNIX computer using something such as adquery or getpwent, you'll see the actual user display name in the GECOS field.

Using the Import from UNIX wizard

After you have created text files with user and group information or verified access to a NIS server and domain, you are ready to perform the first step in the migration process using the Import from UNIX wizard.

To import user and group information:

1. Open Access Manager.

2. Expand Zones and any parent or child zones required to select the zone name into which you want to import users and groups.
3. Select UNIX Data, right-click, then click **Import from UNIX**.
4. Select the import source, then click **Next**.
    - If you select Network Information Service (NIS), type the name of the NIS domain and the host name of the NIS server. The NIS domain and server must be accessible from the Windows network for information to be imported successfully.
    - If you select UNIX configuration files, click **Browse** to locate the text files to import.

    If you selected **Network Information Service** or **UNIX configuration files** in Step 4, go to Step 5.
5. Select the import options you want to use, then click **Next**.

    The import options displayed depend on the import source. For example, if you selected UNIX configuration files and specified a text file containing user accounts and a text file containing group accounts, the import options are:
    - **Include system accounts** to include accounts with UID or GID values from 0 to 99.

      On most computers, accounts with UID or GID values from 0 to 99 are reserved for accounts, such as root, tty, and ftp that you don't need to import or manage using Active Directory. Select the **Include system accounts** option to include these accounts. This option is only displayed if importing from UNIX configuration files.
    - **Automatically shorten the UNIX name to 8 characters** to limit UNIX user and group names to a maximum of 8 characters.

      On some computers, user and group names cannot be longer than 8 characters. If you are importing users and groups that might need access to computers that do not support names longer than 8 characters, you can select **Automatically shorten the Unix name to 8 characters** to automatically truncate the names imported.

      If you are importing from NIS, you can choose to import users, groups, or both.
6. Select a location for storing pending import data, then click **Next**.

    For example, to store pending data for the current zone in an XML file, select **Store in XML file** and specify the location for the file. If the file does not already exist in the default location, you are prompted to create it. To select another location for the XML file, click **Browse**.
7. Review the summary of information to be imported, and select the **Check data conflicts while importing** option if you want to check for conflicts and potential matching candidates during the import process, then click **Finish**.

    If you are importing a large number of users or groups, selecting **Check data conflicts while importing** can cause the import process to take some time to complete. If you don't select this option, you must check the status of users or groups after importing.

After you close the Import from UNIX wizard, users and groups are placed in Active Directory or in an XML file with the status of Pending Import. You must then decide how each user and group should be mapped to accounts in Active Directory.

Checking for conflicts and matching candidates

To move a user or group from Pending Import to a UNIX profile attached to an Active Directory user or group, you must first check for potential conflicts and for potential matching user or group candidates in Active Directory. If you selected the **Check data conflicts while importing** option in the Import from UNIX wizard, you have already completed this step and can continue to Mapping UNIX profiles to Active Directory accounts.

To check the status of pending information:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name into which you imported users and groups.
3. Expand UNIX Data, then expand Groups and Users to see the Pending Import nodes.

    For example, if you imported information for the "Finance" zone, open that zone, expand UNIX Data, then expand Groups and Users.
4. Select Pending Import to display the list of users or groups to be imported.

For example:

Pending import

5. Select all or a subset of pending import users or groups, right-click, then click **Check status**.
    - For pending import groups, a potential match is an Active Directory group with a common name or sAMAccountName that is the same as the pending import group name.
    - For pending import users, a potential match is an Active Directory user with a common name that is the same as the pending import user's GECOS field, or sAMAccountName that is the same as the UNIX user name.

      If there is a match, Access Manager displays that group or user as the default Active Directory candidate and the status as **Ready to import**.

      If Access Manager can't identify a potential match in Active Directory or there are other issues, the status for the pending import group or user describes the issue encountered.

Mapping UNIX profiles to Active Directory accounts

After you check the status of pending import groups or users, you can map the pending import group or user to an Active Directory group or user. The actions you can take depend on the object you select and its current state. For example, if you select a pending group, you can choose to:

- Accept the default Active Directory candidate for the selected group if a candidate is identified.
- Create a new Active Directory group and attach the selected UNIX group profile to it.
- Extend an existing Active Directory group to include the selected UNIX group profile.
- Merge the members of the selected UNIX group with an existing UNIX group in Active Directory.
- Delete the selected UNIX group.
- View and modify the properties of the selected UNIX group.

Accepting the Active Directory candidate

If Access Manager finds a potential match for the pending import group or user in Active Directory, it displays the matching candidate in the details pane. You can accept the suggested candidate by right-clicking the pending import group or user, then selecting **Accept**. After you accept the Active Directory candidate for a pending group or user, the group or user is removed from the Pending Import list.

If all of the pending import group members have an Active Directory candidate associated with them, they are added as members of the Active Directory group. However, the group will remain in the Pending Import list until all of its members are successfully mapped to Active Directory users or removed as members.

Creating a new Active Directory account

If Access Manager did not find a potential match in Active Directory, you must determine whether the pending import group or user should be mapped to an existing Active Directory account or requires a new Active Directory account. If the pending group or user requires a new Active Directory account, right-click the pending group or user, then select the **Create new** option to open the wizard for creating a new Active Directory group or a new Active Directory user.

Follow the prompts displayed in the wizard to provide the additional information needed to create the group or user account.

Adding a profile to an existing Active Directory account

If Access Manager did not find a potential match in Active Directory but an appropriate Active Directory account exists, you must map the pending import group or user to the appropriate Active Directory group or user. If the pending import profile should be added to an existing Active Directory group or user, right-click the pending group or user, then select the **Extend existing** option to open the wizard for adding a UNIX profile to an existing Active Directory group or existing Active Directory user.

Merging pending group members into an existing group

If Access Manager did not find a potential match for a Pending Import group in Active Directory, you might want to merge the members of the Pending Import

group into a group that already has a UNIX profile in the zone. If you want to add the members of a selected pending import group to an existing group profile, right-click the pending import group, then select the **Merge into existing Unix group** option to open the wizard for merging the membership of a pending import group with the membership of an existing UNIX group.

Deleting a UNIX profile for a pending group or user

If there are no suitable candidates to map a pending import group or user, you might want to remove a pending group or user from the Pending Import list. If you want to delete a pending import group or user, you can do so by right-clicking the pending import group or user, then selecting the **Delete** option.

Viewing or modifying properties for a pending group or user

If there are conflicts between a pending import profile and information in Active Directory, you might need to modify the properties associated with the pending import profile before you can take any other action. If you want to view or modify the properties for a pending import group or user, right-click the pending import group or user, then select **Properties**.

If you select a pending group, the properties include the UNIX profile, the time of the import, the file location the information was imported from, the members of the group, and the status of the group.

If you select a pending user, the properties include the UNIX profile, the time of the import, the file location the information was imported from, and the status of the user.

Resolving errors and conflicts

In some cases, you might encounter errors (Error) that must be resolved before a pending import user or group can be migrated into Active Directory. For example, pending import groups cannot be imported if the group profile has any of the following problems:

- The group's GID is negative.
- There is another UNIX group with the same GID already defined in the zone.
- There is a UNIX group with the same group name already defined in the zone.
- The matching Active Directory candidate already has a UNIX profile in the zone.

Similarly, pending import users cannot be imported if the user profile has any of the following problems:

- The user's UID is negative.
- The user's primary group GID is negative.
- There is a UNIX user with the same user name already defined in the zone.

In most cases, you must resolve these issues by modifying the properties for the pending import profile. For example, assume you are importing a passwd file that includes the UNIX user account pierre with the UID 1001, but there is already an UNIX profile in the zone with the UNIX name pierre and UID of 500. After you check the status, the Pending Import list of users will indicate there is an error.

To resolve a conflict like this, you might select the pending import user, right-click, then select **Properties** to change the UNIX user name from pierre to another name, such as pierre2. You should keep in mind, however, that conflicts like this might require investigation to determine the appropriate course of action. For example, if you are attempting to import the UNIX profile for the user pierre and there's a conflict, you need to determine whether pierre with the UID of 1001 is the same person as pierre with a UID of 500 and where each UID is applicable. If both profiles are for one person accessing different computers, you might simply need to define a computer-level override on the specific computer where the UID of 1001 is required. If the pending import user actually refers to a different person, you might have to map the profile to a different Active Directory account or move the computer to a different zone.

Resolving warnings

In addition to the errors that prevent users or groups from being imported, there are several conditions that generate a warning (Warning). Warnings indicate potential problems that you should try to resolve. After you check the status for pending import groups and users, the most common warning is "No matching Active Directory candidate is found." To continue, you must identify or create an Active Directory account for the pending import profile.

If you make changes to a pending import user or group to correct problems, you should click **Check status** after the change to check for any additional issues that might need to be resolved.

Overriding and modifying user properties

If you are using hierarchical zones, user profile information is inherited from parent zones into any child zones you define. You can override the inherited profile attributes at any time to create a new user profile in a specific child zone or on individual computers, if needed. Overriding profile attributes enables you to migrate legacy local accounts without modifying any existing account information or file and directory ownership.

You can also modify either the user profile or the Active Directory user account properties for any user at any time using the tool of your choice. For example, you can use Access Manager, the Access Module for Windows PowerShell, ADEdit, Active Directory Users and Computers, or the Centrify Windows API to modify the zone profile or Active Directory properties for a selected user.

To override a profile attribute in a user profile:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name in which you want to override a profile attribute.

   For example, if you want to override the default login shell in the child zone that only AIX computers join, you might expand Child Zones to view and select the IBM AIX Only zone.

   If you want to override a profile attribute for a specific computer, expand Computers to select the computer name on which you want to override the profile attribute. For example, if you want to override the default numeric identifier for a user on the AIX computer aix6v0.ajax.org, you might expand the IBM AIX Only child zone and the Computers node to view and select the aix6v0.ajax.org computer.
3. Expand UNIX Data for the zone or computer, then select Users.
4. Select the user, right-click, then select **Zone Profile**.

   The profile displays the attributes inherited from the parent zone or currently set.
5. Select an attribute and provide the override value.

   For example, select **Shell** and type /usr/bin/ksh to give the selected user profilea different default login shell—one appropriate for an AIX computer—in the selected zone or on the selected computer.
6. Click **OK** to save the profile change.

Overriding and Modifying Group Properties

If you are using hierarchical zones, group profile information is inherited from parent zones into any child zones you define. You can override the inherited profile attributes at any time to create a new group profile in a specific child zone or on individual computers, if needed.

You can also modify either the group profile or the Active Directory group properties for any group at any time using the tool of your choice. For example, you can use Access Manager, the Access Module for Windows PowerShell, ADEdit, Active Directory Users and Computers, or the Centrify Windows API to modify the zone profile or Active Directory properties for a selected group.

Adding Users or Groups from a Trusted Forest

In most cases, when you create a profile for a user or group in a zone, the Active Directory account already exists in the local Active Directory forest. You can, however, also add profiles for remote users and groups to a zone without adding them to the local forest. If you have established a one-way or two-way trust relationship with a remote or external Active Directory forest, you can add users and groups from that forest to a selected Centrify zone.

You add remote or external users and groups to the zone in the same way you add profiles for local Active Directory users and groups except that you must select the remote forest or domain before searching for the user or group account. For example, at Step 4 of the procedure To create a group profile for an Active Directory group using Access Manager, click **Browse** to select a trusted external forest or a specific domain in the trusted forest.

If you have defined a one-way or two-way trust between a local forest (wonder.land) and a remote forest (w2k3r2.dev), you can select the remote forest in the Browse for container dialog box to add groups from that forest (w2k3r2.dev) to the currently selected zone.

Add groups from the forest

If you use attribute variables to define any part of the user profile, keep in mind that the Centrify Agent cannot directly read any of the attributes for a user from a one-way trusted forest. The agent can retrieve the userPrincipalName and sAMAccountName from the zone profile for the user. However, the agent cannot retrieve other user attributes. If the agent cannot resolve a variable in the user profile, the agent leaves the attribute value undefined. For example, if you use the displayName variable to define the GECOS attribute, that attribute will be undefined for all users from an external forest with a one-way trust.

Identifying users from remote forests

You can identify the Active Directory users who have been added from a remote or external forest by checking the icon displayed in the Access Manager console. If a user is added from a remote or external forest, the user name displays the following icon:

Remote or external forest

Valid login names for users from a remote forest

If you add users from an external forest to a zone, you should be aware that those users can only log on or be identified using the following information:

- A valid UNIX profile name that has a complete set of profile attributes.
- The full Active Directory user name including the user's external forest domain name.

When users are defined in a local forest, they can be located in Active Directory by their UNIX profile name, their userPrincipalName, or their sAMAccountName in the form of their user logon name alone or in the format of domainname\username, so any of these login name formats can be used to access user information or to log on to a Centrify-managed computer.

To identify a user from a trusted external forest, however, you must use either the user's UNIX profile name for the zone or the user's sAMAccountName followed by the user's external domain name in the form of sAMAccountName@domainname. Using the UNIX profile name or the sAMAccountName@domainname ensures the name is unique when there are cross-forest trust relationships. For example, if an Active Directory user from a trusted external forest (sierra.org) has the Active Directory logon name of sofia.perez and a UNIX profile name of sofiapz, the user can be identified using:

- sofia.perez@sierra.org
- sofiapz

You cannot use sierra\sofia.perez or sofia.perez without the domain to retrieve information or authenticate from a remote forest. In addition, the userPrincipalName (username@domainname) for any user might be different from the sAMAccountName@domainname. For example, if you use alternate UPN suffixes, the domain name used in the userPrincipalName might be different from the domain name that uniquely identifies the user. Similarly, a user's logon name (sAMAccountName) might be different from the user name used in the userPrincipalName. For example, if the Active Directory user sofia.perez@sierra.org has a user logon name of SIERRA\perez.s, that user would be found as perez.s@sierra.org.

Adding Multiple Profiles for a User to a Zone

It is possible for a single Active Directory user to have more than one UNIX profile defined in a zone. If you attempt to add a new UNIX profile for an Active Directory account that already has a UNIX profile in the current zone, Access Manager displays a warning but allows you to continue.

If an Active Directory user has more than one UNIX profile in a zone, however, the user should log on to computers in the zone with the UNIX profile name he wants to use. Logging on with the Active Directory user login name—the user's sAMAccountName attribute—might prevent the user from accessing some files because the account has multiple UNIX profiles and UIDs associated with it. In most cases, users can log on with their Active Directory account name if you have created parent and child hierarchical zones that address conflicting profile attributes. However, if you are using classic zones or hierarchical zones that don't address the need for multiple UNIX profiles, users might encounter file ownership issues.

Enabling and Disabling Users in Classic Zones

If you have added user profiles to classic zones, you can enable or disable their UNIX profiles in those zones at any time. Enabling and disabling a UNIX profile is not applicable in hierarchical zones.

To enable or disable the UNIX profile for multiple users in a classic zone, select all of the user names to enable or disable using the CTRL or SHIFT keys, right-click, then click **Enable UNIX Account** or **Disable UNIX Account**.

Forcing Replication for Read-Only Domain Controllers

If the Active Directory forest includes read-only domain controllers, you should force replications when adding or modifying users and groups in a zone. Forcing replication ensures that the new information is available right away.

To force replication after updating a zone:

1. Click Start > Administrative Tools > Active Directory Sites and Services.
2. Expand Sites, then select the Active Directory site that contains the connection over which you want to replicate directory information.

   For example, select **Default-First-Site-Name**.
3. Expand **Servers**, then select the read-only domain controller for which you want to force replication.
4. Click **NTDS Settings**.
5. In the details pane, right-click the connection over which you want to replicate directory information, then click **Replicate Now**.

If you choose not to force replication, the changes made to the zone will not take effect until replication is complete for the forest.

Using Configuration Parameters and Group Policies

You can use local configuration parameters or applied group policies to manage many operations for users and groups on Linux and UNIX computers. For example, you can use configuration parameters or group policies to bypass Active Directory authentication for specific users or to allow some users or groups to be approved for prevalidation. For more information about working with group policies, see the *Group Policy Guide*. For more information about setting parameters in the Centrify configuration file, see the *Configuration and Tuning Reference Guide*.

Enabling and configuring local account management

The local account management features described earlier in this guide require that local account management be enabled and configured.

Several configuration parameters and group policies let you control whether local account management is enabled in your environment, and how local account management is configured after it is enabled.

Local account management is disabled by default unless you are upgrading from a release in which local account management was enabled.

Follow these guidelines to determine whether you need to enable local account management:

- If you perform a fresh installation of Server Suite, the **Enable Local Account Management Feature** group policy is set to **Disabled**, and the adclient.local.account.manage configuration parameter on each local (agent-managed) computer is set to false. To use the local account management features described in this guide, you must manually enable local account management by setting the **Enable Local Account Management Feature** group policy to **Enabled**, or by setting the adclient.local.account.manage configuration parameter to true.

  See the following sections, "Group Policies" and "Configuration Parameters," for more information.

- If you are upgrading from a previous release, you can check the **Enable Local Account Management Feature** group policy setting to enable or disable local account management.

The following information is a summary of how various parameters and group policies affect local account management enablement and configuration. For more details about these parameters and group policies, see the *Configuration and Tuning Reference Guide* and the *Group Policy Guide*.

Group Policies

- **Enable Local Account Management Feature**: Use this group policy to control whether local accounts are managed by the UNIX agent and Access Manager. This group policy is disabled by default, unless you are upgrading from a previous release in which local account management was enabled.

  This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Local Account Management.

  This group policy controls the adclient.local.account.manage configuration parameter.

- **Notification Command Line:** Use this group policy to define a command to process changes to local account profiles after the agent synchronizes local user and group profiles with profiles defined in Access Manager.

This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Local Account Management.

This group policy controls the adclient.local.account.notification.cli configuration parameter.

- **Set refresh interval for access control cache**: Use this group policy to specify how often etc/group and etc/passwd are updated on UNIX and Linux computers, based on the local group and local user settings that you configure in Access Manager. This group policy also controls how often the authorization store cache is updated.

  This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings.

  This group policy controls the adclient.refresh.interval.dz configuration parameter.

Configuration Parameters

The following configuration parameters are located in the /etc/centrifydc/centrifydc.conf configuration file.

- **adclient.local.account.manage**: Use this parameter to control whether local account management is enabled on an individual computer. This parameter has a value of false by default, unless you are upgrading from a previous release in which local account management was enabled.

- **adclient.local.account.notification.cli**: Use this parameter to define a command to process changes to local account profiles after the agent

synchronizes local user and group profiles with profiles defined in Access Manager.

- **adclient.local.account.notification.cli.arg.length.max**: Use this parameter to specify the maximum argument length for the command that you define in the adclient.local.account.notification.cli parameter.
- **adclient.refresh.interval.dz**: Use this parameter to specify how often etc/group and etc/passwd are updated on an individual computer based on the local group and local user settings that you configure in Access Manager. This parameter also controls how often the authorization store cache is updated.

Authorizing Basic Access

This chapter describes the basic principles of authorization and how to grant access to Centrify-managed computers using the default predefined rights and role definitions for Linux and UNIX computers. You should review the information in this chapter before creating custom role-based access rights and role definitions.

Delinea

Basic concepts of Access Rights and Roles

To log on and use Centrify-managed computers, Active Directory users must have a complete UNIX profile and be assigned to at least one role that grants them access. Both the profile and the role assignment can be explicitly defined for the zone or for an individual computer, or inherited from a parent zone.

You can use Access Manager to centrally manage what users can do on computers that have the Centrify Agent installed. For example, you can control who can log on or connect remotely for each computer in a zone through the definition of rights and the assignment of roles. A **right** represents a specific operation that a user is allowed to perform. A **role** is a collection of rights that can be defined in a parent or child zone and assigned to Active Directory users and groups.

The most basic rights are the predefined **system rights** that determine whether a user can log on locally with a password, log on remotely without a password, and run commands in a standard shell or in a restricted shell. The most common settings for these system rights are defined by default in the UNIX Login role so that you can grant users access to Centrifymanaged computers by simply assigning the predefined UNIX Login role and without defining any custom roles or creating any additional access rights.

System Rights Authorize Access in Role Definitions

System rights are always associated with a role definition, whether it is a predefined role such as the UNIX Login role or a custom role you create. You can enable or disable specific system rights in any role definition, but you cannot add, modify, or delete the rights themselves. For Linux and UNIX computers, you can select the following system rights for any role:

- **Password login and non password (SSO) login are allowed**: Specifies that a user is allowed to log on interactively using a password or without a password using a single sign-on token.
- **Non password (SSO) login is allowed**: Specifies that a user is allowed to log on using a single sign-on token.
- **Account disabled in AD can be used by sudo, cron, etc.**: Specifies that an account that is disabled in Active Directory is allowed to access the computer. This right is intended to allow service accounts that run without a password to perform operations.
- **Login with non-Restricted Shell**: Controls whether a user gets a standard shell or is forced into a restricted shell. Users must be assigned at least one role with this right to have access to a standard shell environment. A restricted shell only allows a user to execute explicitly defined commands.

In addition to the platform-specific system rights, there is a system right that allows users to bypass auditing or role restrictions to log on when there are problems on a computer. By selecting the **Rescue rights** option you can allow users in a particular role to log on in situations when all users would normally be locked out. For example, if authentication, authorization information, or auditing is required but not available, most users are prevented from logging on. You can use the rescue rights option to allow selected administrators to access the computer and fix the issues that are preventing other users from logging on.

Note: If you do not explicitly set the **Allow users assigned to this role to log on if problems with authentication, authorization or auditing services prevent logon access** rescue right option for any users, only the local root account will have rescue rights. The root account is always allowed to log on by default.

Access Rights Defined in the UNIX Login Role

The predefined UNIX Login role is configured by default to allow users to log on locally with a password, connect remotely to a computer without being prompted for a password, and access the standard shell environment. The UNIX Login role is also configured to allow users to access all PAM-enabled applications in their environment. The UNIX Login role grants access to PAM-enabled applications through a predefined loginall PAM access right.

For most users and organizations, the default settings in the UNIX Login role make the user experience consistent before and after deploying the Centrify Agent and joining an Active Directory domain. Users can log on and use the shell environment and applications in the same way they did before the deployment of the Centrify Agent.

The predefined UNIX Login role and predefined loginall PAM access right are available by default in every zone. Depending on your requirements and policies, you can assign the UNIX Login role to all Active Directory users or to specific Active Directory users and groups. You can also choose whether to assign the UNIX Login role in parent or child zones to control where different groups of users can log on to Linux and UNIX computers.

Users must have both a complete identity profile and at least one role assignment that grants access before they can log on to any Centrify-managed computer. If you don't use the UNIX Login role, you must create at least one custom role definition that provides similar functionality.

Default Access Rights and Roles

In addition to the predefined UNIX Login role that grants basic access to Centrify-managed computers during deployment, there are other predefined access rights and role definitions that are available by default in every zone. These other predefined rights and role definitions provide specialized access rights for specific scenarios that are common in Linux and UNIX environments.

Default PAM access rights

For Linux and UNIX computers, the following predefined PAM access rights are available:

- login-all grants access to all PAM-enabled applications by specifying the asterisk (*) wild card for the application name. This right is included in the predefined UNIX Login role. You can add this right to any custom role to grant access to all PAM applications, such as login, ftp, ssh, telnet, and many others, without specifying them individually.
- ssh grants access to secure shell sessions on Debian and Ubuntu 6 and 7 computers. By default, this access right grants users access to all secure shell applications and operations.
- sshd grants access to secure shell sessions on all Linux and UNIX computers except Debian and Ubuntu 6 and 7 computers. By default, this access right grants users access to all secure shell applications and operations.

Default secure shell (SSH) access rights

Secure shell (SSH) access rights enable you to limit what users who are granted the PAM ssh or sshd right can do. These rights have no effect without the PAM ssh or sshd right. In addition, the default secure shell rights are only applicable for the Centrify-compiled version of OpenSSH.

For Linux and UNIX computers, the following predefined secure shell access rights are available:

- dzssh-all grants access to all secure shell services.
- dzssh-direct-tcpip allows local and dynamic port forwarding (ssh-L, ssh -D).
- dzssh-exec allows command execution.
- dzssh-scp allows secure copy (scp) operations.
- dzssh-sftp allows secure file transfer (sftp) operations.
- dzssh-shell allows secure terminal (tty/pty) connections.
- dzssh-Subsystem allows an external subsystem except sftp subsystem which has its own right.
- dzssh-tcpip-forward allows remote port forwarding (ssh -R).
- dzssh-tunnel allows tunnel device forwarding.
- dzssh-X11-forwarding allows X11 forwarding.

Predefined role definitions

In addition to the predefined UNIX Login role, there are several predefined role definitions that are available by default in every zone. For Linux and UNIX computers, the following predefined role definitions are available:

- listed makes a user profile visible in a zone but does not grant any type of access rights, PAM rights, or command rights. This is a specialized role that can be used when a user profile must exist for computers in a zone, but no local or remote access should be granted. For example, if a user owning files on a computer in a zone should no longer have access to the computers in the zone, you can assign the listed role so that the files continue to have an owner, but the user has no effective logon rights in the zone.
- local listed makes a local user profile visible in a zone but does not grant any type of access rights. This is a specialized role that can be used when a user profile must exist for computers in a zone, but no user access should be granted. For example, if a user owning files on a computer in a zone should no longer have access to the computers in the zone, you can assign the listed role so that the files continue to have an owner, but the user still has no effective rights in the zone.
- require MFA for login forces two-step authentication for access. This role does not grant access to any PAM applications but can be used in

combination with the UNIX Login role to require users who are assigned to both roles to provide more than one form of authentication. You can also use this role with custom roles that grant access to specific applications if you want to require multi-factor authentication for those applications. You should note that using this predefined role definition requires additional configuration outside of Access Manager. For more information about what is required to support multi-factor authentication, see Requiring multi-factor authentication to log on.

- Rescue - always permit login enables users to log on to computers if there are problems with the authentication, authorization, or auditing service that are preventing other users from logging on. For example, if auditing is required on a computer and the auditing service is not available, only users assigned to a role with the “rescue” system right will be able to log on.
- scp grants secure copy (scp) access rights.
- sftp grants secure file transfer (sftp) access rights.

Identifying the Scope for Role Definitions

The rights from multiple role assignments accumulate, which provides great flexibility and granularity in how you define and assign rights and roles. For example, you can use the UNIX Login role to control basic access, and define a second role that grants the rights to execute a set of privileged commands, so that a user assigned to both roles could log on, but only execute a few specific commands with elevated privileged. By separating rights into separate role definitions, not every role requires PAM applications or system rights, as long as a user is assigned a role that has those rights.

Because access rights are additive, however, it is important to consider where you define and assign roles to control who has administrative privileges on which computers. For example, it might seem reasonable to assign the predefined UNIX Login role to all Active Directory users. Doing so, however, could grant broad permission to log on to Linux or UNIX computers to which you want to restrict access. If you assign that role in a parent zone, it is inherited along with any additional rights granted in child zones.

In most cases, it is appropriate to define roles in parent zones, but assign roles carefully in child zones to avoid granting access rights on computers that host administrative applications or sensitive information.

Assigning the UNIX Login Role

The predefined UNIX Login role allows Active Directory users to log on to Centrify managed computers using any PAMenabled application—such as login, ssh, or ftp—with a default shell and permission to execute the same set of commands available to any standard UNIX user account. By default, the UNIX Login role is configured to take effect immediately and never expire. By default, the UNIX Login role is also configured to audit user activity if the auditing service is running on a computer users access.

The default settings are appropriate for most Linux and UNIX users in most organizations. However, you can change any of the default settings in either a parent or a child zone, if needed.

What to do before assigning the UNIX Login role

You can assign the UNIX Login role to all Active Directory users, to specific Active Directory users, or to specific Active Directory groups. Because the UNIX Login role is a predefined role, you cannot assign any local users to the role.

Before you assign the role, you should decide whether you want to assign and inherit the role from a parent zone or make the assignment in a specific child zone. You should also decide whether you want to specify optional start and end times for some role assignments.

Rights required for this task

The following table describes the minimum rights that must be granted for users to successfully manage role assignments in a zone:

Authorization	On the **Object** tab, select **Allow** for the following: List contents Read all properties Create all child objects Delete all child objects On the **Properties** tab, select **Allow** for the following: Write msDS-AzApplicationData	This object only
	On the **Properties** tab, select **Allow** for the following: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole	The msDS-AzRole object
AzRoleObjectContainer	On the **Object** tab, select **Allow** for the following: List contents Read all properties Create msDS-AzRole objects Delete msDS-AzRole objects	The msDS-AzApplication object and all child objects
	On the **Properties** tab, select **Allow** for the following: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole	The msDS-AzRole object
	On the **Properties** tab, select **Allow** for the following: Write msDS-AzApplicationData	The msDS-AzAdminManager object
AzOpObjectContainer	On the **Object** tab, select **Allow** for the following: Read all properties Create msDS-AzOperation objects Delete msDS-AzOperation objects Create msDS-AzRole objects Delete msDS-AzRole objects	This object only
	On the **Properties** tab, select **Allow** for the following properties: Write displayName Write msDS-AzApplicationData Write msDS-TasksForAzRole Write msDS-MembersForAzRole	The msDS-AzRole object
	On the **Properties** tab, select **Allow** for the following: Read name Read Name Write msDS-AzApplicationData Write name Write description	The msDS-AzOperation object

Who should perform this task

A UNIX administrator who manages one or more zones most often performs this task, depending on your organization's policies.

How often you should perform this task

In most organizations, you assign the UNIX Login role to target groups of users at a time during deployment and as needed, thereafter.

Steps for completing this task

The following instructions illustrate how to assign the UNIX Login role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To assign users and groups to the UNIX Login role in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to make role assignments.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the UNIX Login role definition from the list of roles, then click **OK**.

   By default, the role is set to start immediately and never expire. You can set a **Start time**, **End time**, or both start and end times for the role assignment. For example, if the role assignment applies to a contractor who will be hired for a specific period of time and you want to automatically disable the role after they finish the job and leave the organization, you can specify the start and end times when you assign the role.
6. Select whether the role assignment applies to all Active Directory accounts or specific accounts.

   If you want to automatically assign the role to every user added to the Active Directory forest or trusted forests, you can select **All Active Directory accounts** for convenience. This option is similar to selecting the "Authenticated Users" or "Everyone" system groups. For example, if you want to assign all Active Directory users the UNIX Login role by default, you can select this option. Only users who also have a complete UNIX profile will be able to log on to the UNIX computers joined to the domain.

   If you are assigning the role to specific accounts, click **Add AD Account** to search for and select the Active Directory groups or users to assign to the role, then click **OK**.
7. Click **OK** to complete the role assignment.

What to do next

Verify Active Directory users or group members assigned the UNIX Login role can log on to Centrify-managed computers in the zone where you have made the role assignment.

Where you can find additional information

If you want to learn more about working with rights, roles, and role assignments, see the following topics for additional information:

- Defining rights to use commands
- Defining rights to use PAM applications
- Using secure shell session-based rights
- Creating and assigning custom role definitions

Performing Role Assignment on Multiple Computers

To simplify the process of assigning Active Directory users or groups to a role, you can perform a bulk role assignment. With a bulk role assignment, you can assign a role to multiple Active Directory users and groups on multiple computers at the same time. For example, if you have two groups of Oracle administrators and three computers where the members of those groups need access to their OracleAdmin role, you can select those two groups and those three computers to be assigned the OracleAdmin role in the same process. You can also specify optional start and end times for the role assignment and have those settings apply for all of the users, groups, and computers you have selected for bulk assignment.

To assign a role to multiple users and groups on multiple computers

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to make a bulk role assignment.
3. Right-click, then select **Assign Roles to Computers**.
4. Type the user and group names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for user and group names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria. If multiple users or groups match the search criteria, select the appropriate users and groups, then click **OK**.
5. Type the computer names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for the computer names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria. If multiple computers match the search criteria, select the appropriate computers, then click **OK**.
6. Select a role for the list of roles available, then click **OK**.
7. Review the role assignment start and end time and the user and group accounts that are being assigned the role, then click **OK**.

   You can make changes to the start and end times if you want those changes applied for all of the users, groups, and computers that are part of this bulk role assignment.

After you click OK, the selected users and groups are then automatically assigned the selected role on the selected computers.

Viewing Rights and Roles

Access Manager allows you to view the status and effective rights for any Active Directory user or local user in a zone, whether they have been assigned a role or not. You can view detailed information about the rights and role assignments for users by using **Show Effective UNIX User Rights**. If a user is not assigned a role or does not have a complete user profile, be certain to select the **Show omitted users** option, otherwise, information will not be shown for the user.

> **Note:** Local users are defined in Access Manager in the zone and are saved in /etc/passwd on each computer in each zone where the profile is defined. Local users that you define in the zone do not need to be Active Directory users. For more information about local users, including information that is required for a user profile to be complete, see Creating user profiles.

To view rights for an individual user in Access Manager

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to view rights and other account details.
3. Right-click, then select **Show Effective UNIX User Rights**.
4. Select a computer or click Browse if you want to limit the information included to a specific computer.
5. Select **Show AD users** and **Show local users** as necessary, depending on which users you want to view. One or both of these choices might already be selected, depending on the location from which you originally selected **Show Effective UNIX User** rights.
6. Select **Show omitted users** to include users who have an incomplete profile or do not have a role assignment in the list of UNIX users.

   User information is displayed as shown in the following example. Key points about the information displayed are as follows:

   - Users with incomplete profiles are displayed in red (if **Show omitted users** is selected).
   - Local users are not required to have an AD name, resulting in a displayed **AD Name** value of N/A.
   - AD users are not required to have a UNIX profile, resulting in a displayed **Profile State** value of N/A.
   - For more information about the differences between AD users and local users, as well as details about profile states for local users, see Creating, modifying, and deleting user profiles for local users.

     Creating, modifying, and deleting user profiles
7. Select a user to see more detailed information about the user's profile, role assignments, and rights in the selected zone or on a specific computer:
   - Click **Zone Profile** to review the UNIX profile defined for a user and where the profile attributes are defined. If a user has an incomplete profile, you can click the **Zone Profile** tab to see which profile attributes are missing.
   - Click **Role Assignments** to review a user's role assignments. The Object Assigned column indicates whether the role is explicitly assigned to the user (user@domain) or to a group the user is a member of (group@domain). The Location of Assignment column indicates the zone or computer role in which the assignment was made. Information for the Start Time, End Time, or both columns is only displayed if a role assignment has time constraints.
   - Click **PAM Accesses** to review the PAM application access rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.
   - Click **Commands** to review the command access rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.
   - Click **SSH Rights** to review the secure shell rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.
8. Click **Close** when you are finished reviewing user rights in a zone or on particular computers.

Checking rights and roles with the dzinfo program

You can also view rights and roles for specific users or the current user by running the dzinfo command-line program on Centrifymanaged computers. If you want to use the dzinfo program to view roles and rights for other users, however, you must have root permission.

You can run dzinfo without any arguments to see your own rights and role assignments. The command displays detailed information about the your role assignments, the availability for each role assignments, your effective rights, the current audit level, and the specific PAM access, command, and secure shell rights you have been granted.

To see more detailed information, such as the days and times a role is available, you can use the --verbose option. For example, to see detailed information, you could type the following command:

dzinfo --verbose

To view roles and rights for a specific user:

1. Log on or switch to root on a managed computer.
2. Run the dzinfo command for a specific user with the username in the command line.

   dzinfo username

   For example, to see details about the rights and roles assigned to the user sonya, you could type the following command:

   dzinfo sonya

If rights and role assignments have been configured for the specified user, the command displays detailed information about the user's role assignments, the availability of those role assignments, the user's effective rights, the audit level in effect, and the specific rights that have been granted.

You can also use the dzinfo program to test whether a user has the right to run specific commands. For more information about using dzinfo and the dzinfo command line options, see the dzinfo man page.

Changing the Audit Level for Role Definitions

By default, all role definitions—including predefined role definitions—are set to "Audit if possible" as the audit level. With this setting, user activity is audited if the auditing service is installed and enabled on a managed computer. If the auditing service is not installed or not running on a given computer, this setting has no effect. Users can log on and use the access rights that are defined for their role assignment without having their activity audited.

In most cases, the default "Audit if possible" setting is appropriate because it doesn't block user access if you are not deploying the auditing infrastructure but will automatically capture user activity if you are deploying auditing. In some cases, however, you might want to change the audit level. You can modify the audit level for any role definition to specify whether users must be audited in order to log on.

To change the audit level for a role definition:

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to change the audit level.
3. Expand Authorization and Role Definitions.
4. Select a role definition, right-click, then select **Properties**.
5. Click the **Audit** tab.
6. Select the appropriate audit level to use for the role definition.
    - Select **Audit not requested/required** if you are not interested in auditing session activity for users in the role.
    - Select **Audit if possible** if you want to audit user activity on computers running the Centrify auditing service. If you select this option and the auditing service is not installed or not currently available, users assigned to the role are allowed to log on without having their activity audited. This option is selected by default for new roles.
    - Select **Audit required** if you want to audit all session activity for users assigned to the role. If you select this option and the auditing service is not installed or not currently available, users assigned to this role are not allowed to log on.

      If auditing is required for users in a role, you should also define a role with rescue rights to allow selected administrators to log on and correct problems when other users are locked out. For more information about creating a role with rescue rights, see Creating a role definition with rescue rights.
7. Click **OK** to save the role definition.

Delinea

Requiring Multi-Factor Authentication to Log On

You can configure multi-factor authentication for users logging on to Centrify-managed computers to improve the security of physical or virtual data centers. You can assign the predefined require MFA for login role in combination with the UNIX Login role to require users who are assigned to both roles to provide more than one form of authentication. You can also create custom role definitions with the **Require multifactor authentication for login** system right. Before setting this system right, however, you should be aware the multifactor authentication for Centrify-managed computers relies on the infrastructure provided by the Centrify Identity Platform.

As a preview, here are the steps involved to enable multi-factor authentication for Centrifymanaged computers in hierarchical zones:

- Register for Privileged Access Service.
- Install and configure at least one **connector** for communication with Privileged Access Service.
- Verify the users who are required to provide more than one form of authentication have valid **Active Directory accounts** that are active in Privileged Access Service.
- Add or select the **authentication profiles** that specify the types of authentication challenges to support.
- Create a role with the appropriate **computer members and administrative rights** for multifactor authentication.
- Verify the **identity platform instance URL** you want to use if you have access to more than one instance.

After you have completed the preliminary steps, you can assign users the predefined require MFA for login role or a custom role with the **Require multi-factor authentication for login** system right to require two-step authentication when logging on using PAM applications. These preliminary steps are also required if you want to create command rights that require two-step authentication when executing commands using elevated privileges (dzdo) or in restricted shell (dzsh) environments.

The preliminary steps are also required to support multi-factor authentication in classic zone and Auto Zone. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication.

For more information, see the Preparing to use multi-factor authentication

Defining Rights to Use Commands

As discussed in Basic concepts of access rights and roles, access rights allow users to perform specific operations. You define the most basic rights—such as the right to log on or connect remotely—when you define roles. However, you can use more granular command access rights to tightly control who has access to individual command-line programs. This chapter describes how to define access rights that allow users to execute command-line programs on Centrify-managed computers.

Controlling Access to Commands

In a standard UNIX shell environment, an ordinary user account can execute a large number of common command-line programs without any special privileges, and one or more administrative accounts, such as root, are required to execute commands that perform privileged operations. If ordinary users need to execute any of the commands requiring administrative privileges, they might have to switch to an administrative account that requires them to know the password for a privileged users or been granted access by configuration settings in a sudoers file.

For Centrify-managed Linux and UNIX computers, however, you can define command access rights to tightly control the specific commands users can execute. You can also refine those rights to only allow specific arguments to be used or to require an executable to be located in a specific directory.

There are no predefined rights for commands. Therefore, only the specific command access rights you define will be available for you to add to roles. You should keep in mind that any command rights you define are specific to the zone where you configure them, but can be used in any child zones of that zone.

What Command Rights Provide

Command access rights identify the specific commands that can be executed on a Linux or UNIX computer by a user assigned the role to which the rights are added. Command rights also specify whether the commands defined in the right are executed under the user's own account or using another user account.

There are two primary reasons for defining command rights:

- To **grant access** to specific commands that must be executed with elevated privileges
- To **restrict access** to only allow specific commands to be executed.

Granting access using command rights

The most common reason for defining a command right is to grant access to commands that perform privileged operations. For example, you might want to grant users additional privileges to execute specific commands in a standard shell environment that they are not otherwise allowed to execute with the default rights associated with their account.

With this type of command right, most commands are executed in the default shell environment with ordinary user privileges. When users assigned to a role with this type of command right want to use their elevated privileges, they invoke the command they have been granted access to using the dzdo command. This type of command right is similar to configuring privileges in a sudoers file, then invoking a command using sudo.

This type of command right is appropriate for UNIX users who have a standard shell environment and only need elevated rights to perform specific tasks.

Restricting access using command rights

It is less common, but also possible to define a command right to restrict access. For example, you might want to create a role that provides strictly controlled access to an explicitly defined subset of shell commands. This type of command right creates a customized restricted environment shell (dzsh) where only explicitly defined commands can be executed. This type of command right is similar to configuring a "whitelist" of allowed command and is appropriate for users who only need access to a limited set of commands to perform their job.

Controlling the Shell Environment for Commands

You can define command rights to control who has permission to run specific commands in a zone. When you define individual command rights, you can also specify whether the commands can be executed in a non-restricted shell environment, a restricted shell environment, or both. After you define the command right, you can then add it to an appropriate role definition. It is then the role definition to which you add the command right that controls whether users can use the command in a standard, unrestricted shell environment or in a restricted shell environment.

If the role definition allows a non-restricted shell environment—like the UNIX Login role—the command right provides functionality similar to the UNIX sudo command except that it uses the role settings and the zone authorization store rather than through a sudoers configuration file.

If the role definition does not allow access to a non-restricted shell environment, the command right can only be used in a restricted shell environment and users assigned to the role can only execute the specific commands explicitly defined in command right.

Defining Rights to Run Privileged Commands

The most common reason for creating a command right is to allow users to execute commands that require privileges not granted to a standard UNIX user account. For example, you might want to grant some users permission to run Centrify command-line programs that require root privileges to better manage their own computers.

Defining command rights that grant elevated privileges is similar to granting access to privileged commands using the sudoers configuration file and the sudo program.

Steps for completing this task

The following instructions illustrate how to define a command right to execute a command with elevated privileges. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a command right for privileged access

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to define a command right.
3. Expand Authorization and UNIX Right Definitions, then select Commands.
4. Right-click, then click **New Command**.
5. On the General tab, type a short descriptive name for the command right, and optionally, a more detailed description for the command right.

   The privileged command name is required and must not be more than 63 characters in length or contain any special characters, such as asterisks (*), slashes (\ /), question marks (?), or quotation marks (").
6. Type the command you want to add.

   The Command field is required and should include any parameters or options, if needed. You can also use wild cards or a regular expression to specify commands matching a particular pattern.
7. Select the type of pattern matching to use for the "Command" and "Specific path" fields.
   - Select **Glob expressions** to use glob pattern matching syntax for wild cards.
   - Select **Regular expressions** to use extended regular expression pattern matching.

     For more information about pattern matching, see Selecting the pattern matching syntax.
8. Select an appropriate path for matching the command on the different operating environments you support.
   - Select **Standard user path** to use the local operating system's common set of user directories to find the command.
   - Select **Standard system path** to use the directories the root user would normally get on the local operating environment to find the command.
   - Select **System search path** if you want to search for the command in a predefined set of locations. The search locations are defined using the dzdo.search_path configuration parameter. If you select **System search path** and the dzdo.search_path parameter is not defined, the current user's path is used to search for the command.
   - Select **Specific path** if you want to define a custom set of locations for finding the command specified. If you select this option, you can specify one or more paths, separated by a colon.

     If you are specifying a path, the path must start with a forward slash (/) unless you are matching all paths (*). For example, if the command you specify is ls and you set the path to *, the ls command from any path is allowed.

     If you set both the "Command" field and the "Specific path" field to match all strings (*), any command from any path is allowed.
9. Specify an integer that determines the priority of the command — the lower the number, the higher the priority.

   If there are multiple commands that match the pattern you specified for the "Command" field, the priority determines which command has higher

priority.

10. Click the **Run As** tab, then select **Can be used by dzdo** to allow the command to be added to a role for privileged execution.
11. Select the user or group accounts that can be used to execute the command.
    - Select **Any User** if any standard user account can be used to execute the command with dzdo.
    - Select **One of the following users, uids, groups or gids** if you want to specify one or more user or groups that can be used to execute the command with dzdo.

      In most cases, the local root account is the appropriate account to use because it allows ordinary users to execute the specified command using root account privileges. However, you can click **Add** to add other users, groups, or service accounts that can be used to execute the command. Use the format #UID for UID values, %group for group names, or %#GID for GID values.

      The account used to execute commands can be an Active Directory user with a UNIX profile in the zone or a local UNIX user account. However, the account used to log on and invokes the command using dzdo must be associated with an Active Directory account.

      Optionally, you can specify the primary groups can be used when executing the command using dzdo:
    - Select **Any Group** if any group can be used as the primary group when executing the command with dzdo.
    - Select **One of the following groups**, then click **Add** if you want to specify the groups that can be used as the primary group when executing this command with dzdo.

      You can also configure commands to be executed using dzdo in a restricted shell environment. For this example, however, the command right is only used in a nonrestricted shell environment.
12. Click **OK** to save the new command right.

    In most cases, you can use the default settings for environment variables and execution attributes.
    - If you want to keep, remove, or add environment variables for command execution, see Customizing environment variables for command execution.
    - If you want customize any of the execution attributes, see Customizing command execution attributes.

Creating a role to run commands with elevated privileges

On most Linux and UNIX computers, you can identify commands that require elevated permissions, who can run those commands, and where different users or groups can run the commands using a sudoers configuration file. Users who have been granted the appropriate permissions can run privileged commands by invoking the sudo command.

Centrify provides similar functionality, but the commands are configured by defining command rights, adding the rights to the appropriate roles, and assigning the roles to different users and groups. Users who have been assigned the appropriate roles can then run privileged commands by invoking the dzdo command.

If users are assigned the predefined UNIX Login role, they have access to all of the standard command-line programs that are available to ordinary UNIX users. You can create a separate role for commands that run using root or another privileged user account. Alternatively, you can combine command rights and system rights in a custom role definition or by adding the command rights to the default UNIX Login role.

Command rights that allow users to execute commands with elevated privileges should only be added to roles with the **Login with Non-Restricted Shell** system right.

Users must execute command rights that grant elevated privileges using the dzdo command. If you selected the **Re-authenticate current user** option as an execution attribute when defining a command right, users must also provide the password for their own account, their own password and one or more other forms of authentication, or the types of authentication determined by the authentication profile configured in Privileged Access Service, which might or might not involve providing a password.

If you selected the **Re-authenticate using the target user's password** option as an execution attribute when defining a command right, users must also provide the password for the account used to execute the command.

To create a role that can execute commands with elevated privileges, do the following:

- Create command rights for the privileged commands users are allowed to run.
- Create a new role definition and set the System Rights for the role to allow password login, nonpassword login, or both, and select the **Login with Non-Restricted Shell** option, then click **OK** to save the role definition.
- Right-click the role, select **Add Right**, then select login-all or a specific PAM access right and the privileges command rights users are allowed to run, then click **OK** to save the changes to the role definition.

For more information about creating, assigning, and testing custom role definitions, see Customizing command execution attributes.

Defining a Restricted Shell Command Right

You can also use command rights to strictly control which commands certain users can execute. In a restricted shell environment, users can only execute the specific commands and command-line options that are explicitly allowed. For example, you might want to grant some users permission to run a specific Centrify command-line program, such as adinfo, without allowing them to run any other command-line programs on some computers.

Users who are assigned to a role with the restricted shell environment are not be able to run any other commands, including informational commands such ls, ps, and whoami, unless you explicitly include them in the command right. You are not required to explicitly add basic navigational commands, such as cd and pwd, to the command right.

What the restricted shell provides

For Linux and UNIX computers, Centrify provides a customized Bourne shell, dzsh, to serve as the restricted shell environment. The dzsh restricted shell supports environment variables, job control, command history, and the specific command rights you define. For example, you can use the up-arrow key in the dzsh shell to recall previously-entered commands. You can also set a limit to the command history available by adding HISTSIZE=n to the $HOME/.dzshrc file.

For most operations, working in the dzsh shell is similar to working in an unrestricted shell except that the command set available is limited to the command rights you add to the environment.

Limitations of the restricted shell

The restricted shell environment does not enforce rights for commands that run outside of the shell. For example, if users run a graphical desktop manager, they can run commands and applications that are launched from menu selections in the graphical user interface.

In addition, the command rights defined for the dzsh shell do not prevent users from running built-in shell commands, accessing the file system, or seeing process or system information. For example, even in a restricted shell environment with no rights to run any commands, users in a dzsh shell could get a process listing using the following script:

```
for i in /proc/[0-9]*;
do read PROC < $i/cmdline;
echo $PROC;
done
```

Because the shell scripting environment allows the operations, users can effectively access information that the commands defined for the restricted shell environment do not allow.

Securing the restricted shell environment

There are many ways sophisticated users can get around limitations placed on a restricted shell environment. For example, most text editors, such as vi and emacs, allow shell escapes. Giving users permission to run programs that allow shell escapes in a restricted shell enables them to open a new unrestricted shell environment with none of the restrictions placed on them in their defined environment, Similarly, giving users access to commands that set or modify local time and date settings might allow users to avoid time constraints for running commands or the expiration date and time for specific role assignments.

In some cases, even individual command line options might provide users with the means to run commands not defined in their restricted shell environment. For example, defining a command right that allows users to run the tar command with the usecompressprogram program_name option allows user to run the specified program_name even though the program_name is not an allowed command in their restricted shell environment.

In choosing the commands to allow in a restricted shell, therefore, you should carefully consider ways to plug potential security holes the commands might introduce or whether there are alternative commands that provide the same functionality more securely. For example, if you need to give a user access to an editor, such as vi or vim, you could restrict the ability to execute nested commands to prevent users from opening a new shell from within the editor. Alternatively, you could add the rvi command to the restricted environment instead of vi or vim because rvi doesn't allow the user to open a new shell.

For more information about setting attributes that control command executions, see Customizing command execution attributes.

Steps for completing this task

The following instructions illustrate how to define a command right for use in a restricted shell using Access Manager. For more information about any step, see Defining rights to run privileged commands. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a command right for restricted shell access

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to define a command right.
3. Expand Authorization and UNIX Right Definitions, then select Commands.
4. Right-click, then click **New Command**.
5. Type a short descriptive name for the command right, and optionally, a more detailed description for the command right.
6. Type the command you want to add.
7. Select the type of pattern matching to use for the "Command" and "Specific path" fields.
8. Select an appropriate path for matching the command on the different operating environments you support.
9. Specify an integer that determines the priority of the command—the lower the number, the higher the priority.
10. Click the **Restricted Shell** tab, then select **Can be used in a restricted role** to allow the command to be added to a role that runs in a restricted shell environment.
11. Select whether commands are executed using the user's logon account or using a specific the user name or UID.

    If you want to configure commands to be executed using dzdo in a restricted shell environment, you can click the Run As tab to specify a user or group for command execution.
12. Click **OK** to save the new command right.

    In most cases, you can use the default settings for environment variables and execution attributes.

    - If you want to keep, remove, or add environment variables for command execution, see Customizing environment variables for command execution.
    - If you want customize any of the execution attributes, see Customizing command execution attributes.

Creating a role to run commands in a restricted shell

For Linux and UNIX computers, Centrify provides a customized Bourne shell, dzsh, to serve as a restricted shell environment. The dzsh restricted shell supports environment variables, job control, command history, and the command access rights you define.

To create a role that runs a restricted shell, do the following:

- Create command rights for the restricted shell commands users are allowed to run.
- Create a new role definition and set the System Rights for the role to allow password login, nonpassword login, or both, and verify that the **Login with Non-Restricted Shell** option is not selected, then click **OK** to save the role definition.
- Right-click the role, select **Add Right**, then select login-all or a specific PAM access right and the restricted shell command rights users are allowed to run, then click **OK** to save the changes to the role definition.

For more information about creating, assigning, and testing custom role definitions, see Customizing command execution attributes.

Selecting the Pattern Matching Syntax

When you define a command right, you can specify whether you want to use glob pattern matching syntax or extended regular expression syntax to match the strings specified for the “Command” and “Specific path” fields.

Customizing Environment Variables for Command Execution

You can customize the environment variables used during command execution in both the non-restricted and restricted shell environments. For example, if a command is executed using a specific user or service account that requires environment variables that are not defined for the user invoking a command, you can define those environment variables as part of the command right definition.

If you want to configure the environment variables to use for a command right, click the **Environment** tab. You can then select one of the following options:

- Reset environment variables
- Remove unsafe environment variables
- Add environment variables

Resetting environment variables

Select **Reset environment variables** if you want to define the list of environment variables to set when the user runs the command. Note that only the environment variables you explicitly specify are retained and those environment variables will replace the default set of environment variables, rather than append the default set of environment variables. You can use Access Manager or dzdo.env_* configuration parameters in the centrifydc.conf file to control the list of environment variables to use when executing commands. For example, you can set the dzdo.env_keep configuration parameter in the centrifydc.conf file to keep a specific set of environment variables like this:

dzdo.env_keep: VAR

With this setting, only the VAR environment variable is defined for the list of environment variables to keep. All other environment variables, including the default list of user environment variables—such as PATH and KRB5CCNAME—are removed.

If you select this option, click **Edit** to specify the environment variables to retain from the user's environment in a comma-separated list. Click **Add**, type the environment variable name, then click **OK** for each environment variable you want to retain.

Removing environment variables

Select **Remove unsafe environment variables** if you want to remove a specific set of unsafe environment variables when the user runs the command. The list of unsafe environment variables is defined by the dzdo.env_delete configuration parameter in the centrifydc.conf file. Note that only the environment variables you explicitly specify are removed.

If you select this option, click **Edit** to specify the environment variables to remove from the user's environment in a comma-separated list. Click **Add**, type the environment variable name, then click **OK** for each environment variable you want to remove.

Adding environment variables

Select **Add environment variables** to define new environment variables to add when the user runs the command. Enter variables in a comma-separated list in the form name=value, or click **Edit** then **Add** to add new variables and values. You can add new variables regardless of which of the other options you select.

Customizing Command Execution Attributes

You can modify the default command execution attributes that are used when commands run in either the non-restricted shell or in a restricted shell environment. In most cases, changes are rarely required for commands that run in a non-restricted shell. It is more common to change the execution attributes for commands that run in restricted shell environments. For example, you can use the execution attributes to control whether an allowed command can invoke a nested command. In a restricted shell environment, you might want to prevent a command from invoking nested commands to reduce the chance that users can run commands not explicitly defined for their environment.

If you want to set any execution attributes for a command right, click the **Attributes** tab. You can then select different options to control different aspects of command execution.

Requiring re-authentication to run commands

After successful authentication during the login process, you can control whether running a command in a restricted shell or using elevated privileges requires re-authentication or not. If you want to require re-authentication, select the authentication rules to apply. When defining the rights for executing commands, you can select from the following authentication options:

- No re-authentication required

  Select this option to allow users to run the command without any additional authentication.

- Re-authenticate current user

  Select this option to require the user to be re-authenticated before running the command using their own credentials. If you select this option, you can also specify whether reauthentication requires the user to provide their password, requires their password and another form of authentication, or requires multi-factor authentication as determined by the authentication profile configured in Privileged Access Service, which might or might not involve providing a password.

  If you select both **Use password** and **Require multi-factor authentication for login,** users are prompted to type their password and provide another form of authentication before the command is executed. If you have configured the authentication profile to accept more than one type of authentication challenge, users are prompted to select the authentication method to continue.

- Re-authenticate using the target user's password.

  Select this option to require the user to be re-authenticated before running the command using the target run-as user's credentials.

Preserving group membership

When defining command rights, you should consider whether keeping a user's existing group membership would provide benefits for command execution or could be exploited to perform unauthorized operations.Select **Preserve group membership** if you want to retain the logged-on user's group membership while executing commands.

Allowing nested commands

When defining command rights, you should consider whether allowing the execution of nested commands could be exploited to perform unauthorized operations. Select **Allow nested command execution** if you want to allow a command to invoke another program or open a new shell. To enhance the security of a restricted shell environment, you should deselect this option to prevent an allowed command to be used to run another program or open an unrestricted shell.

Preventing unsafe path navigation

When defining command rights, you should consider whether the command or any of the allowed command arguments could be exploited to perform unauthorized operations. One way command arguments can be exploited is to allow navigation up the path hierarchy. To prevent command arguments from allowing unsafe navigation up a path hierarchy, you can select the **Prevent navigation up a path hierarchy**. For example, if a command right allows a user to execute a command such as vi /etc/httpd/conf/* without this option, the right could be exploited by specifying a command argument that navigates up a path hierarchy to perform an unauthorized operation. In this case, the right might be used to edit any file as the root user by specifying a relative path as a command-line argument.

vi /etc/httpd/conf/../../shadowpass

You can avoid this potential security risk by disabling upward path navigation for command arguments, if needed. Note that this setting is only supported in

hierarchical zones and is only applicable for glob command rights.

Setting the umask value

Set the **Umask value** by selecting the read (R), write (W), and execute (X) permissions for the owner, group, and other users if you want to change the permission settings for executing a command.

Setting SELinux role-based access control

Configure the **SELinux Setting** for dzdo Security Enhanced Linux (SELinux) role-based access control (RBAC). By enabling the SELinux role and SELinux type fields, privileged commands can be specified with the default role and type for creating SELinux context in execution. These settings can be overridden using the '-r'/'-t' command-line options respectively. To enable this setting, click the **SELinux Setting** button and enable SELinux role and SELinux type, then enter string values in the corresponding text fields. Settings are saved in the attribute of the msDS-AzOperation command object.

Note: These settings are currently supported only on the RHEL systems and effective only on system with SELinux enabled and joined to a hierarchical zone.

Setting the command digest

You can use Digest Settings to specify SHA-2 digests so that sudo can verify the binary's checksum (SHA-2) before sudo executes the binary. The supported digest (hash) types are as follows:

- SHA224
- SHA256
- SHA384
- SHA512

Select a digest type, and then enter a checksum. You can specify multiple digests for a command.

Note that setting a command digest is only supported in the explicit path matches against the command right, and only supported in the hierarchical zone.

Testing Command Rights

After command rights have been defined, added to role definitions, and assigned, you can use dzinfo --test "command" to check whether you have permission to execute a specific command. You can use the dzinfo username --test "command" command to check whether a specified user has permission to run a specified command. If you want to use the dzinfo program to view command rights for other users, however, you must have root permission.

To check for command rights, you must enter the complete path to the command and enclose the command in single or double quotes. For example, to test whether the user, qa1 has a command right that allows execution of the id command as root, you could run the following command on a Linux or UNIX computer:

[user1@rh5]# dzinfo qa1 --test "/bin/id"

Depending on the role definition and the user's role assignment, the command might display information similar to this:

Testing: User = qa1 command = /bin/id

User qa1 can run the command as 'root' via dzdo, authentication will not be required, noexec mode is off

Using Command Rights in a Standard Shell

After command rights have been defined, added to role definitions, and assigned, users can execute privileged commands in a standard shell environment by invoking the dzdo command then typing the command to execute, including any command-line options they are allowed to use.

For example, assume you have defines a command right for shutdown -r that enables users to execute the command as the root user. If you add that right to a role definition—such as the UNIX Login role—that allows users to log on using a standard shell environment, users assigned to that role can execute the command by typing the following:

```
dzdo shutdown -r
```

Using Command Rights in a Restricted Shell Environment

After command rights have been defined, added to role definitions, and assigned, users can execute commands in a restricted shell environment by typing the command, including any command-line options they are allowed to use.

For example, assume you have defines a command right for shutdown -r that enables users to execute the command as the root user. If you add that right to a role definition that forces users into a restricted shell environment, users assigned to that role can execute the command by typing the following:

shutdown -r

Users can only execute the specific command rights that have been added to the role within the restricted shell environment.

Running unauthorized commands

When users are assigned to roles that require a restricted shell environment, the dzsh shell provides the subset of commands the user is allowed to run and automatically runs each allowed command as the user the command is configured to run as. If the user attempts to run a command he is not authorized to use in his current role, the shell displays a warning. For example, if the user is not authorized to run the uname command in the dzsh shell, the following message is displayed:

$ uname
uname: command not allowed

Setting or changing the active role

Users who are only assigned to one or more restricted shell environments roles are only allowed to run commands within the dzsh shell. Within the restricted shell, a user can only be in one active role at a time to prevent ambiguity about the commands the user can run or the user account that should be used to execute those commands.

For example, if the user carol is assigned to the lab_staff restricted shell environment role that specifies the tar command should run as root and to the temps restricted shell environment role that specifies the tar command should run as tmp_admin, she needs to specify which role she is using to run the tar commands under the proper account.

Within the restricted shell, users can switch between available roles, as needed, using the built-in role command. If a user has been assigned to the backup_ops role and the dev_managers role, he can run the role command to specify which role should be active so that only commands from that role apply. For example, to switch from the backup_ops role to the dev_managers role:

$ role dev_managers
Role changed to: dev_managers

For more information about using the role option in a restricted shell, see the man page for dzsh.

Viewing available roles

The dzinfo command enables users to view information about the roles they have available and what they are allowed to do within their different roles. You may want to add this command to all of your restricted environment roles to allow users to check their definitions and availability within the authentication and privilege elevation restricted environment shell.

For more information about using the dzinfo command, see the man page for dzinfo.

Using a graphical desktop manager in a restricted environment

In some operating environments, users who are placed into a restricted environment may not be able to log on using a graphical user interface desktop manager unless they are explicitly given permission to run the desktop manager or related commands within the dzsh restricted environment. For example, on Red Hat Linux, users must be allowed to run /usr/bin/dbus-launch to log on using KDE or Gnome desktop manager.

To allow restricted environment users to log on using KDE or Gnome on Red Hat, you must add dbus-launch to the list of allowed commands for the restricted environment user's role. If you want to prevent restricted environment users from logging on using the graphical user interface, you can restrict their access to specific PAM-enabled applications such as ssh.

Defining rights to use PAM applications

As discussed in Basic concepts of access rights and roles, access rights allow users to perform specific operations. You define the most basic rights—such as the right to log on or connect remotely—when you define roles. To use the rights associated with a role, however, you must be able to authenticate your identity through a pluggable authentication module (PAM) application, such as login or ssh. This chapter describes how to define PAM application rights that authorize users to log on or access services on Centrify-managed computers.

How applications determine access rights

Most of the programs you run on Linux and UNIX computers are configured to use a pluggable authentication module (PAM) to control access. For example, the login, secure shell (ssh), and file transfer (ftp) services are all PAM-enabled programs. These programs check the local PAM configuration to determine whether a user is allowed to use the requested service.

When you install the Centrify Agent and join a domain, you replace the default PAM authentication service with a PAM service that looks for the users and groups to allow or deny access to in Active Directory. Because the PAM service is the first “gatekeeper” to access on most computers, users must have at least one PAM access right to log on at all.

Default PAM access rights

By default, Access Manager creates three predefined PAM access rights in every parent and child zone:

login-all	All PAM applications on a computer joined to the domain. Adding the login_all PAM access right allows users to log on and use any PAM-enabled application. The right uses the wild card (*) character to match all PAM application names and is included by default in the predefined UNIX Login role.
ssh	Secure shell sessions on Debian and Ubuntu 6 and 7. Adding the ssh PAM access right allows users to log on remotely using secure shell connections on Debian and Ubuntu computers joined to the domain.
sshd	Secure shell sessions on all Linux and UNIX computers except Debian and Ubuntu 6 and 7. Adding the sshd PAM access right allows users to log on remotely using secure shell connections on all other distributions of Linux and UNIX computers joined to the domain.

Adding Specific PAM Access Rights

PAM access rights control who can access specific PAM-enabled applications in the zone where they are created and any child zones of that zone. You can add as many **PAM Access** rights as you need to identify the specific PAM-enabled applications users can access. For example, you can add PAM access rights to control who can use file transfer protocol (ftp) services on specific computers.

If you want to grant rights to specific PAM applications, however, you must know the appropriate application name on the specific computers you support. For example, if you want to allow Active Directory users to log on and use a default shell, you might create a PAM access right for the login program and for a graphical desktop manager such as gdm.

What to do before creating a new access right

Before creating a new PAM access right, you should review the operating system of the computers in the zone where you plan to create the new right. The application name might be different on computers with different operating systems. If you are creating separate rights for individual PAM applications, keep in mind that users must have at least one PAM access right or they will not be able to log on to any computers.

Rights required for this task

You can create new PAM access rights if you have been delegated the "Manage roles and rights" administrative task in the Zone Delegation Wizard. If you have not been delegated this task, your user account must be a domain user with the following permissions:

Authorization	Click the **Properties** tab, then select **Allow** for the following properties: Write msDS-AzApplicationData
msDS-OpObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: Create msDS-AzOperation objects Click the **Properties** tab, then select **Allow** for the following properties: Read objectClass

Who should perform this task

In most cases, a UNIX administrator or a delegated zone administrator familiar with PAM applications and the operating system of the managed computers performs this task, depending on your organization's policies.

How often you should perform this task

It is common to add new PAM access rights over time as the need arises and as you develop more granular control over the specific rights different users should be granted.

Steps for completing this task

The following instructions illustrate how to add a PAM access right using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new PAM access right.
3. Expand Authorization, then expand **UNIX Right Definitions**.
4. Select PAM Access, right-click, then click **Add PAM Access Right**.
5. Type a name for the access right.

   The name of the access right can be the same as the PAM application name, or any name that is easily identifiable.

6. Type the name of the PAM-enabled application for which you want to create an access right.

   You can use wildcards to perform pattern matching for the application name. For example, you can specify *ftp* to match all PAM-enabled applications containing the string ftp, such as vsftpd, ftpd, and ftp.

   The Application Name field supports glob pattern matching syntax. For example, the name can contain a question mark (?) to represent any single character, an asterisk (*) to represent any string, including an empty string, or an expression enclosed by brackets ([. . .]). For more detailed information about using wildcard patterns and glob syntax, see the glob man page.

   You should note that application names vary depending on the local operating system where the application is accessed. For example, the following table lists several common PAM-enabled applications and the appropriate application name to use on different platforms.

telnet	Common Linux platforms, such as Red Hat, Debian, SuSE, Centos, and Ubuntu, HP-UX, and Irix	login
	Sun Solaris	telnet
	VMware ESX, Oracle Linux, Scientific Linux	remote
ftp	Common Linux platforms, such as Red Hat, Oracle Linux, and Scientific Linux, and VMware ESX	vsftpd
	Some Linux platforms, such as Debian, Centos, and Ubuntu, Sun Solaris, HP-UX, Irix	ftp
graphical desktop	Common Linux platforms, such as Red Hat, Debian, Oracle Linux, Centos, Scientific Linux, and Ubuntu	gdm
	Sun Solaris and HP-UX	dtlogin
	SuSE and Irix	xdm
ssh	Most platforms	sshd
	Debain and Ubuntu	ssh

1. Type an optional description of the access right.
2. Click **OK** to save the PAM access right.

What to do next

After you define a new PAM access right, you might want to create a new role definition and add this right to it in the current zone or in a child zone. You must add the right to a role to test its operation.

Modifying an existing PAM access right

After you have created and tested a new PAM access right, you might want to modify the right name and description, or the pattern used to match the application name. For example, if you add computers with a different operating system to the zone where the PAM access right is defined, you might have to modify the application name to use wild card characters.

To modify a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to modify, right-click, then click **Properties**.
5. Change the right name, application name, or description for the access right, then click **OK**.

Copying a PAM access right

You should keep in mind that PAM access rights are specific to the zone where you create them. They can be added to any roles you define for the zone or to roles defined in any child zone of the zone. After you define PAM access rights in a zone, however, you can also copy and paste or drag and drop the rights from one zone to another, as needed.

To copy a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to copy, right-click, then click **Copy**.
5. Navigate to the PAM Access node in the new zone, right-click, then click **Paste**.

Deleting a PAM access right

If you are no longer using a specific PAM access right in any roles, you might want to delete the right. For example, if you create new rights for testing purposes and found some of them were not appropriate or failed to work as expected, you might want to delete the rights you aren't going to use.

To delete a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to delete, right-click, then click **Delete**.

Renaming a PAM access right

If you only need to change the name of a PAM access right, you can rename the right at any time without modifying the description or the pattern used to match the application name.

To rename a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to rename, right-click, then click **Rename** and type a new name for the access right.

Using PAM-enabled applications

The default UNIX Login role includes the predefined login-all PAM access right to enable users to log on and access any PAM-enabled application. If you define specific PAM access rights, users who are assigned a role with that right can only access the specifically authorized PAM-enabled applications. For example, users who are assigned to a role that includes the right to access FTP (ftpd) can connect to the FTP server by typing a command similar to the following:

ftp ginger.ajax.org

Requiring multi-factor authentication for PAM applications

If you select the "Multi-factor authentication required" system right in a role definition, the PAM applications you add to the role will require users to select a secondary form of authentication to log on successfully. You define the forms of authentication available and presented to the user in the authentication profile you have configured using the administrative portal for the Centrify Platform. For example, you might configure an authentication profile that require users to answer a phone call, click a link in an email message, or respond to a text message.

Note that some applications do not support multi-factor authentication and users might be denied access to applications that they would otherwise be able to use. For example, if a specific version of an application that you want to use only supports a single layer of authentication—such as a password challenge—users would be prevented from logging on and using the service even if they are assigned to a role with the predefined login-all PAM application right.

If you want to grant access to applications that only support one layer of authentication in roles where you are generally using the "Multi-factor authentication required" system right, you must add those applications to the list of applications for which you want to skip multifactor authentication. You can update the list of applications for which to skip multifactor authentication by enabling and modifying the "Specify programs for which multi-factor authentication is ignored" group policy or setting the pam.mfa.program.ignore configuration parameter in the centrifydc.conf file.

Before assigning roles with multi-factor authentication required to users, you should test access to all of the applications you expect users to access to verify they won't be unexpectedly denied access simply because multi-factor authentication isn't supported. Because the applications that don't support multi-factor authentication will depend on the platforms and the versions of the applications you plan to support, testing in your own environment is the only way to determine which applications to add to the pam.mfa.program.ignore configuration parameter.

The most common applications that are known to only support a single password challenge and response for authentication are ignored for multi-factor authentication by default. For example, some versions of java and vsftpd do not support multi-factor authentication and are ignored by default.

Additionally, while some platforms support multi-factor authentication for all PAM applications, they may not allow you to require multi-factor authentication for GUI log in. For example, for users running AIX, Solaris, and HP-UX, multi-factor authentication for GUI login is not supported.

Options applied to the Centrify PAM module

The authentication service applies some options to the Centrify PAM module pam_centrifydc module. For example, in a RHEL system, you would see the following in the /etc/pam.d/system-auth file:

auth sufficient pam_centrifydc.so

auth requisite pam_centrifydc.so deny

account sufficient pam_centrifydc.so

account requisite pam_centrifydc.so deny

session required pam_centrifydc.so homedir

password sufficient pam_centrifydc.so try_first_pass

password requisite pam_centrifydc.so deny

In addition, you could see the following in the /etc/pam.d/su file:

auth sufficient pam_centrifydc.so enable_dzpamgate

For each management group, a set or stack of modules can be defined and used in turn. When an application calls the PAM library function (for example to authenticate), the PAM runtime will call each authentication function in each module— one at a time like cards from a stack. The order of calling is determined by the order in the configuration (service) file.

Be careful when changing the order in the stack; changing the order might have impact on the functionality considerably.

There are four types of PAM services:

- Authentication service modules
- Account management modules
- Session management modules

- Password management modules

There are four control flags:

- **Requisite**: The requisite flag is probably the strongest of the flags. If a module is flagged as requisite, and it fails (returns not-OK), PAM will return to the calling application instantly and report the failure.
- **Required**: The return code for a required module is stored. In the case of failure, execution is not stopped but continues to the next module. When the stack of modules has been executed, and at least one required module has failed, PAM will return failure to the calling application. Moreover, the failure is associated with the first failing module) The required control flag is useful in keeping unauthorized persons out of your computer, particularly since the other modules in the stack are applied as well.
- **Sufficient**: A sufficient module can actually be quite strong. The processing of the stack is stopped if a sufficient module returns OK, if no previous required module has failed. If there are required modules after the sufficient modules, these modules are not called.
- **Optional**: When a module is flagged as optional, a failure does not alter the execution of the stack as in the case of the requisite flag. Moreover, the return code is ignored, and neither failure nor success is taken into account.

The list of options that are applied to pam_centrifydc are as follows:

- Deny: When it gets to this line, it will just deny the request
- Try_first_pass, use_first_pass, get_first_pass: All three follow PAM standards.
- Try_first_pass: Use the password from the previous stacked authentication module, and prompt for a new password if the retrieved password is blank or incorrect.
- Use_first_pass: The default is to use the old password saved by a previous module, or if none, to ask for it. With use_first_pass it fails if there is no old password.
- Requisite: This is an HP platform only option. The effect is the same as HP "pam requisite"
- Unix_cred: This is Solaris only option to provide Solaris's pam_unix_cred
- Homedir: Create user home directory
- Enable_dzpamgate: This is a Centrify option to seal a potential pam security hole. In some systems, pam account module does not always gets called so the authorization checking in auth module is inserted with the enable_dzpamdate option.

In /etc/centrifydc/centrifydc.conf, the parameter equivalents are provided as:

pam_mkhomedir.so [umask=mode] [skel=skeldir]

pam.homedir.perms: 0700

pam.homeskel.dir: /etc/skel

To update the parameters, remove the # sign and change values as desired.

Using secure shell session-based rights

As discussed in Default secure shell (SSH) access rights, **Access Manager includes predefined secure shell rights that enable you to** identify the specific secure shell services that a user who has the PAM SSH access right can run. This chapter describes how to use the secure shell (ssh) session-based rights that control the operations specific users or groups can perform on Centrifymanaged computers.

Secure shell rights require Centrify OpenSSH

SSH has become the defacto standard for administrators and users to securely access remote UNIX systems. The combination of the latest versions of OpenSSH supporting Kerberized connections, along with the DirectControl Agent directly integrating the UNIX computer with Active Directory's Kerberos infrastructure, provides the administrator with the ideal environment for secured single sign-on. Users logging in from Windows computers can securely access remote UNIX computers using their Active Directory credentials to automatically log in to the UNIX computer.

While many UNIX systems might have an sshd server installed, most are older implementations of the sshd server that do not support Kerberos and newer versions might not have been compiled with support for Kerberos. The Centrify package contains OpenSSH compiled with support for Kerberos by dynamically linking to the Centrify Kerberos libraries to ensure that single sign-on works seamlessly as expected in an Active Directory environment.

This provides several advantages, including:

- The OpenSSH client and server are preconfigured to automatically support PAM and Kerberos.
- There is no need for DNS-to-realm mapping because DirectControl knows the relationship between hosts and their SPNs.
- There is no need for a .k5login file in the user's home directory since DirectControl can automatically map the UPN (User Principal Name) in the Kerberos ticket to the UNIX profile for the Active Directory username presented in the ticket.
- OpenSSH in combination with DirectControl accepts connections to any of the computer's valid hostnames, either fully qualified or not, because all combinations are registered with Active Directory. This further reduces the dependency on accurate DNS entries to enable Kerberos to operate properly.
- The installation process automatically updates the $PATH.

The Centrify version of OpenSSH is a separate package that can be installed with the Centrify Agent. Before you configure any specific secure shell rights to include in roles, verify that you have the Centrify OpenSSH package installed on your managed computers. The default secure shell rights are only applicable for the Centrify-compiled version of OpenSSH. If you did not select the OpenSSH package as part of a custom installation when you installed the agent, re-run the installation script to install the package before attempting to use secure shell rights.

Secure shell rights require PAM access rights

Before you configure any specific secure shell rights, you should also identify the PAM access right to use. The predefined PAM access right sshd—or ssh for Ubuntu computers—grants users permission to log on and use all secure shell services on Centrify-managed computers. You must grant the sshd, ssh, login-all, or a custom PAM access right before you can use any secure shell (SSH) rights to restrict access to specific services.

The SSH access rights only work in conjunction with the PAM access right that allows a user to log on using a secure shell session. If a user is not assigned to a role that grants the PAM access right to log on using a secure shell, SSH rights are ignored.

When a user attempts to log on using a secure shell session, adclient first verifies that at least one role in effect for the user has the PAM access right that allows him too log on using SSH. If a PAM access right is in effect, adclient checks to see which specific SSH rights the user has before allowing or denying the action the user is attempting.

Combining secure shell rights

You can add predefined SSH rights to any role that can be assigned to Active Directory users and can combine different rights for fine-grain control over the specific secure shell operations users are allowed to perform. For Linux and UNIX computers, only the following predefined secure shell session-based rights are available:

- dzssh-all grants access to all secure shell services.
- dzssh-direct-tcpip allows local and dynamic port forwarding (ssh-L, ssh -D).
- dzssh-exec allows command execution.
- dzssh-scp allows secure copy (scp) operations.
- dzssh-sftp allows secure file transfer (sftp) operations.
- dzssh-shell allows secure terminal (tty/pty) connections.
- dzssh-Subsystem allows an external subsystem except sftp subsystem which has its own right.
- dzssh-tcpip-forward allows remote port forwarding (ssh -R).
- dzssh-tunnel allows tunnel device forwarding.
- dzssh-X11-forwarding allows X11 forwarding.

When combining rights into role definitions, you should keep in mind that some secure shell operations require you to explicitly include the dzssh-exec right. For example, if you include the dzssh-scp right in a role definition, a user might attempt to execute an arbitrary program with a command line similar to following:

ssh troll@localhost scp -S/home/troll/script " -f "

Because this command line presents a potential security risk, the operation is not allowed. To prevent the dzssh-scp right from being used on its own to execute an arbitrary program on a remote computer, the -S command line option is only supported if you also include the dzssh-exec right in the role definition. Similarly, you must explicitly include the dzsshexec right in a role definition if you want to support using the dzssh-sftp right with the -S command line option. For security reasons, only the dzsshexec right allows the remote execution of a program on a target computer.

If the dzssh-exec right is not included in the role definition when it is required, users will see an “access denied” message.

You should note that you cannot add any secure shell rights to role definitions that allow local users. You can only include them in role definitions for Active Directory users.

Configuring secure shell settings

You can use Centrify group policies to manage several aspects of secure shell (ssh) authentication and operation. The Centrify group policies for secure shell are located in the **SSH Settings** folder after you add the centrify_unix_settings.xml administrative template to a Group Policy Object. When you enable and configure secure shell group policies, the changes are recorded in the secure shell configuration file, /etc/centrifydc/ssh/sshd_config, at the next group policy update interval. To have your changes take effect immediately, run the adgpupdate command.

Centrify puts all of the configuration files for secure shell operations in the /etc/centrifydc/ssh directory. Depending on your operating system, you might also have other ssh configuration files stored in the other locations. When users start a secure shell session and use their secure shell rights, the Centrify Agent first checks the /etc/centrifydc/ssh directory for configuration files, then looks for configuration file in the /usr/local/etc directory on AIX computers, and in /etc/ssh directory on most other Linux and UNIX computers.

At a minimum, you should enable the **Enable application rights** group policy in a Group Policy Object that applies to the site, domain, or organizational unit that contains Centrify-managed Linux and UNIX computers.

To configure the secure shell group policy for application rights

1. On a Windows computer, open the Group Policy Management console.
2. Select an appropriate Group Policy Object, right-click, then select Edit.

   You can select any Group Policy Object that applies to the site, domain, or organizational unit that contains Centrify-managed Linux and UNIX computers.
3. Expand Computer Configuration > Policies > Centrify Settings > SSH Settings and double-click **Enable application rights**.
4. Click Enable, then click OK.

   This setting adds the following parameter to the /etc/centrifydc/ssh/sshd_config file:

   ServiceAuthLocation /usr/share/centrifydc/libexec/dzsshchk

   This parameter sets the path to the dzsshchk command. The dzsshchk command verifies the access rights for users when they log in with SSH for all computers to which the group policy object applies.

You can also use secure shell group policies to control other configuration settings, such as the allowed and denied groups and users and authentication processing. For example, you can use the following group policies to configure operations for Centrify OpenSSH connections:

- **Add sshd_config properties** enables you configure secure shell properties defined in the sshd_config file by group policy. If you enable this group policy, you can add and edit properties as name-value pairs.
- **Allow challenge-response authentication** enables you use multi-factor authentication if you are using the secure shell package installed with the operating system. This group policy is not required if you are using the Centrify OpenSSH package for the agent.
- **Allow groups** specifies the list of groups whose members are allowed to log on through sshd.
- **Allow GSSAPI authentication** enables authentication either as the result of a successful key exchange, or through GSSAPI user authentication.
- **Allow GSSAPI key exchange** enables authentication using a key exchange based on GSSAPI.
- **Allow users** specifies the list of users who are allowed to log on through sshd.
- **Deny groups** specifies the list of groups whose members are not allowed to log on through sshd.
- **Deny users** specifies the list of users who are not allowed to log on through sshd.
- **Enable application rights** allows secure shell applications to grant secure shell rights.
- **Enable PAM authentication** to use PAM account and session handling.
- **Permit root login** specifies whether the root account can be used to log in using ssh.
- **Set banner path** specifies the path to a local file that is sent to a remote user requesting authentication.
- **Specify authorized keyfile** specifies the file that contains the public keys that can be used for user authentication.

- **Specify ciphers allowed for protocol version 2** enables you to add or delete ciphers allowed for single sign-on connections.
- **Specify client alive interval** specifies a timeout interval, in seconds, for requesting a response to client alive messages.
- **Specify log level** specifies the level of detail to record in the log file for messages from sshd.
- **Specify login grace period** specifies the time, in seconds, after which the server disconnects if a user has failed to log in.
- **Specify maximum client alive count** specifies the maximum number of client alive messages that may be sent by the secure shell daemon (sshd) without receiving a response from the client.

For more information about adding administrative templates for group policies to a Group Policy Object and how to configure and apply the group policies for secure shell, see the *Group Policy Guide*.

Configuring secure shell parameters

The following parameters apply to specific usage for SSH.

ServiceAuthLocation

Uncomment this line in sshd_config to enable the SSH application right feature. Refer to the pre-defined scp, sftp, and winscp for how to utilize this feature. Default is disable.

AuditSshCommandline

Set this parameter in sshd_config to yes if the command line options are displayed in the audit trail message. Default is no.

krb5ccUnique

Set this parameter to yes to specify when storing the Kerberos credentials cache. Centrify sshd generates a unique credential cache name for it. If this parameter is set to no, the old style credential cache name, krb5cc_<uid> or KCM:<uid>, is used.

SSOMFA

Set this parameter is yes to support Single Sign-On (SSO) with Multi- Factor Authentication (MFA). The MFA order is determined by the AuthenticationMethods keyword in sshd_config. This keyword works only when UsePAM is enabled and the Centrify keyword ServiceAuthLocation is set. Default is no.

Note: MFA is not supported for authentication using public key.

Creating and assigning custom role definitions

Access rights and role definitions are intended to give you maximum flexibility to grant or restrict access for the users and groups in your organization. This chapter describes how you can configure role-based access controls for Centrify-managed computers by adding custom access rights to custom role definitions and assigning those custom role definitions to users and groups. This chapter provides examples to illustrate how you can create and assign custom role definitions. The authorization scenarios you can support might be far more complex than the examples described in this guide.

Combining rights into role definitions

Rights can be combined in a variety of ways to accomplish different goals. In general, however, role definitions fall into one of these broad categories:

- Roles that grant access to one or more PAM applications and a standard UNIX shell.

  With this type of role, Active Directory users can log on using all or a specified PAM application, such as login or ftp, and execute commands that are commonly available to non-administrative users. This type of role can only be assigned to Active Directory users or groups.

- Roles that grant users additional privileges to execute administrative commands and perform administrative tasks they would not be able to perform with a standard user account.

  With this type of role, users can temporarily elevated their privileges to execute administrative commands by first invoking the dzdo command, which is similar to sudo. This type of role can be assigned to Active Directory users or to local users.

- Roles that provide access to a specific subset of shell commands in a customized restricted environment shell (dzsh).

  With this type of role, users can execute the commands explicitly defined for them in a restricted shell environment. This type of role can be assigned to Active Directory users or to local users.

In preparing role definitions for different groups of users, you should keep in mind that the rights from multiple role assignments accumulate. For example, you could use one role definition to control login rights, and another role definition to specify a set of privileged commands. By separating login rights from privileged access rights, not every role definition requires PAM application or UNIX system rights.

Creating a root-equivalent role definition

Most organizations require at least one root user role definition that is equivalent to specifying ALL:ALL in a sudoers file or giving users access to the root password on their computers. The purpose of this role definition is to allow selected users to execute privileged commands on a regular basis. The role definition allows them to execute commands without being given the root password or having privileges hard-coded in individual sudoers files on multiple computers.

Because this role definition enables system administrators to execute privileged commands without the root password, you can improve security for the organization and reduce the chance of an audit finding for access to the root password.

You can create this role definition in a parent zone or a child zone to control its scope. In most cases, you should only assign the role in a child zone or on an individual computers.

Define the right for running all commands

Rights and roles are defined at the zone level and inherited down the zone hierarchy. If you define a right in the top-level zone, it is available in all child zones. If you define a right in a child zone, it can be used in that zone and any of its child zones. Similarly, you can define roles in the top-level parent or any child zone, depending on where you want to make the role available. In this example, the right to run all commands as the root user is defined in a toplevel parent zone.

The following instructions illustrate how to define a right for running all commands using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a right for running all commands as root:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.

   For this example, select the top-level parent zone so that this command right is available in all child zones.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a description for this right, then define the right to run all commands like this:
   - Type an asterisk (*) in the Command field to indicate all commands are allowed.
   - Select Specific path and type an asterisk (*) in the field to indicate that any path is allowed.
6. Click the Restricted Shell tab and deselect the **Can be used in a restricted role** option if you want to prevent this command from being used in a role that uses a restricted shell environment.
7. Click the Run As tab to verify the command can be used with dzdo and is set to run as root by default.
8. Click **OK** to use the default environment variable settings and command attributes.

   Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Create a role definition for running all commands

After you have defined the right to allow a user to run any command with root privileges, you can create a role definition for that right. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

To create a role definition with the right to run all commands as root:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.

5. Type a name and description for the new role, then click **OK**.

   For example, type a name such as root_equivalent and descriptive text such as Users with this role can run any command with root privileges.

   Optionally, you can select **Allow local accounts to be assigned to this role** if you want to assign both Active Directory users and local users to the role. This option is only available when you first create a role definition. You can also click **Available Times** if you want to limit when the role is available for use. By default, roles are available at all times.

   If you using the UNIX Login role to grant access to computers in the zone and want to use the default auditing level of **Audit if possible**, you can click **OK** then skip to Step 8.

6. If you are not assigning the UNIX Login role to grant access to computers, click the System Rights tab and select the following options:
   - Password login and non-password (SSO) login are allowed
   - Non-password (SSO) login is allowed
   - Login with non-Restricted Shell

     Note that you cannot set these system rights if you selected the option to allow local users to be assigned to this role.

7. If you don't want to use the default auditing level, click the Audit tab.
   - Select **Audit not requested/required** if you have the auditing service enabled but don't want to audit user activity when this role is used.
   - Select **Audit if possible** to audit user activity where you have the auditing service enabled.
   - Select **Audit required** to always audit user activity. If the auditing service is not available, users in this role are not allowed to log on.

8. Select the new role definition, right-click, then click **Add Right**.
9. Select the right you defined for running all commands as root, then click **OK**.

Assign an Active Directory group to the role

You should associate Centrify role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership. For example, create an Active Directory group named sanfrancisco_role_rootequivalent. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for root-level access, such as root_equivalent, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role.
   - Select Group as the object to find.
   - Optionally, type all or part of the group name.
   - Click Find Now,

     Select the group you created for the role in the results, then click **OK**.

7. Click **OK** to complete the assignment.

Creating a role definition for a shared service account

The root-equivalent role definition provides centralized management for a limited number of administrators who have permission to execute all commands on selected computers. Another common reason for defining a role is to execute privileged commands associated with a service account. In many organizations, service account passwords are known by multiple users, making them a security risk. For example, all of the database administrators in the organization might know the password for an oracle service account, an account with permission to perform privileged database operations. Because the password is shared information, it presents a security risk and a potential audit finding that might have costly consequences.

Setting up a role definition for a service account involves creating a command right for switching to the service account user and defining a PAM access right for role.

Define the right for switching to a service account

The steps for defining a right for switching to the service account user are similar to defining the rights for the root-equivalent user, but the definition is more restrictive.

To define a right for switching to a service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a description for this right, then define the right to switch to the service account.For example, if the service account is oracle:
    - Type su - oracle in the Command field.
    - Verify the Standard user path is selected.
6. Click the Restricted Shell tab, under Can be used in a restricted role, select **Specific user or uid**, then type root.
7. Click the Run As tab, deselect **Can be used by dzdo**.

    These settings specify that this right can only be used in a restricted shell environment and users can only run the commands that are explicitly allowed in the restricted role they are assigned. If this is the only right defined for a role, the only command users assigned to the role can run is su - oracle. For a role definition with this right to be effective, you would add command rights for the specific database operations users should be allowed to perform after switching to the oracle service account. For example, if the oracle service account is used to run a backup-all-dbs script, you would add a right to allow the execution of that script.
8. Click **OK** to use the default environment variable settings and command attributes.

    Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Define a PAM access right to allow logging on

The default UNIX Login role allows users to log on using a password or without a password in an unrestricted environment. If you are creating a role for a service account, you can use PAM access rights to control the specific commands users can use to log in. To illustrate controlling how users log on, this example of a restricted role for the oracle service account only allows users to log on with ssh.

To define a PAM access right for a specific application:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new PAM access right.
3. Expand Authorization > UNIX Right Definitions.
4. Select PAM Access, right-click, then click **Add PAM Access Right**.
5. Type a name and, optionally, a description of the PAM application for which you are adding an access right.

For the Application field, type the platform-specific name for the PAM application as defined in the PAM configuration file or PAM directory. For example, type ssh or sshd. You can also use wildcards in this field to perform pattern matching for the application name.

6. Click **OK** to save the access right for this PAM-enabled application.

Create a restricted role definition for the service account

After you have defined the rights that allow a user to log on using a PAM-enabled application and run the su - command for a service account, you can create a role definition for these rights. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

To create a restricted role definition for switching to a shared service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role, then click **OK**.

   For example, type a name such as oracle_service and descriptive text such as Users with this role can start a secure shell session and switch to oracle.

   By default, this role is available at all times. You can click **Available Times** if you want to specify days of the week or select times of the day for making the role available.
6. Click the System Rights tab and select at least one option that allow users assigned to this role definition to log on, then click **OK**.

   In this example, users open a secure shell to switch to the service account so you might select **Non-password (SSO) login is allowed**.

   If a service account instead of a user account is used to log on, it might be mapped to a disabled Active Directory account. In this case, you might select the **Account disabled in AD can be used by sudo, cron etc** system right to ignore the disabled state and allow the service account to log on.
7. Select the new role definition, right-click, then click **Add Right**.
8. Select the rights you defined for running the switch user (su -) command and logging on with the PAM application ssh, then click **OK**.

Assign an Active Directory group to the role

You should associate Centrify role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership. For example, create an Active Directory group named sanfrancisco_role_oracle. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for using secure shell and switching to the service account access, such as oracle_service, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role definition.
   - Select Group as the object to find.
   - Optionally, type all or part of the group name.
   - Click Find Now.

     Select the group you created for the role in the results, then click **OK**.

7. Click **OK** to complete the assignment.

Working in a restricted shell environment

When users who are assigned to this role want to open a secure shell session and switch to the oracle service account, they will be placed in a restricted shell environment. Within the restricted shell, they can only execute the commands you have added to the role definition until they exit the restricted shell session. In this example, the role definition only allows users to log on using ssh and execute one command, su - oracle. If those users are also assigned the UNIX Login role, they will have access to an unrestricted shell when they close the restricted shell session.

If you want users who access a shared service account to work exclusively within the restricted shell environment, you must remove the UNIX Login role assignment in the zone or on the computer where they should only have restricted shell access. Before removing the UNIX Login role assignment, however, you should consider the trade-off between improved operational security and audit compliance and reduced operational access. Depending on the rights you add to a role that runs in a restricted shell environment, the restricted shell can dramatically limit what users can do.

Testing access in a restricted shell

If you create a role definition for a shared service account that runs in a restricted shell environment, you should test it before migrating any users to it. You can use the dzinfo command with the --test option from a UNIX command prompt. For example, type dzinfo, the user name to test, the --test option, then the full path to the command to test:

dzinfo raejames --test "/usr/bin/su - oracle

You can also run the dzinfo command with the --roles option to see information about the rights defined for the current user or a specified user. For example, run the following command to check the roles and rights defined for the user raejames:

dzinfo raejames --roles

For more information about using this command, see the dzinfo man page.

What users see in a restricted shell environment

For users assigned to a role that runs in a restricted shell, logging on opens a dzsh shell. Within that shell users can only execute the commands you have explicitly defined for them. In this example scenario for a shared service account, typing su - oracle is the only allowed command. If the user types any other command, the shell reports that the command is not allowed.

Define a command that allows root access

The steps for defining a right for switching to the root user are similar to defining the right to run commands for the root-equivalent user, but Centrify recommends you create a separate right definition for this case.

To create the right to switch to the root user:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name, such as emergency_access, for this command right and, optionally, a description for this right, then define the right to switch to the root user:
    - Type the command for switching to the root user. For example, type su - root in the Command field.
    - Verify Standard user path is selected.
6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

    These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.
7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

    In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.

Create a role definition for temporarily running as root

After you have defined the right to switch to the root user, you can create a role definition for that right.

To create a role definition with the right to run the emergency_access command:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role.

    For example, type a name such as emergency_access and descriptive text such as Users with this role can temporarily run commands with root privileges.
6. Click **Available Times** to specify days of the week or select times of the day for making the role definition available.

    For example, you might want to allow access only on Friday, Saturday, and Sunday and deny access the rest of the week. After you have set the days and times for the role definition to be available, click **OK**.
7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select the emergency_access command you defined for switching to the root user, then click **OK**.

    To use this role, a user must be assigned to the UNIX Login role for the zone or a role definition that has at least one UNIX system right, such as Password login and nonpassword (SSO) login are allowed.

Assign the role as a computer-level override

In most cases, a role definition of this type is assigned to a specific computer rather than applied to all computers in a zone.

To make a role assignment on an individual computer:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer for which you want to define a computer-level role assignment.
3. Expand Computers, then select the specific computer on which you want to make a role assignment.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for temporary root access, such as emergency_access, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory user who should have temporary root access:
    - Leave User as the object to find.
    - Optionally, type all or part of the use name.
    - Click Find Now.

      Select the user in the results, then click **OK**.
7. Deselect **Start immediately** and set a specific Start time for the role assignment.
8. Deselect **Never expire** and set a specific End time for the role assignment.
9. Click **OK**.

Verify the role assignment on the computer

You can run dzinfo --roles or dzinfo username --roles to see if the emergency_access role is available based on the start time for the role definition and the local time of the Linux or UNIX computer.

At the specified start time for the role assignment on the local computer, the user you assigned to the emergency_access role can type the following command:

dzdo su - root

The user is not prompted to provide the password and becomes the root user on the local computer until the specified role assignment end time. The one caveat to be aware of is that the user would continue to have root access after the specified end time if the shell session remains open continuously. If a user is still logged on after the time period has expired, you should check whether the user still requires root-level access. If the session has remained open but the user should no longer have root access, kill the session and log the user off.

Creating a role definition with specific privileges

The previous examples of role definitions granted broad privileges. You can also use role definitions grant or deny very specific rights. For example, you might want to deny access to a specific set of commands for a specific group of administrators who otherwise have broad access rights or to strictly limit exactly what commands users can execute. Depending on the requirements of your organization, you might configure these types of role definitions to be used in a restricted or unrestricted shell.

The steps for creating a role definition with specific privileges are similar to the steps for creating the other roles. In this example, rights are defined to prevent the execution of specific commands and combined with a right to grant access to all commands not explicitly listed.

Define command rights to prevent the use of commands

The steps for defining rights that deny access to specific commands are similar to the steps defining other rights, but require different syntax. In this example, you create a “blacklist” of commands users cannot execute.

To create the right to switch to the root user:

1. Open Access Manager.

2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name, such as No password resets, for this command right and, optionally, a description for this right, then define the right:
    - Type !passwd * in the Command field.
    - Verify Standard user path is selected.

    An exclamation point (!) at the start of a command disallows matching commands. Command rights that start with the exclamation point take precedence over others that don't.
6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

    These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.
7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

    In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.
8. Repeat Step 4 to Step 7 to create rights for the following specific commands:

    !groupadd *
    !useradd *
    !groupdel *
    !userdel *

Create a restricted shell role definition that uses the command rights

After you have defined all of the command rights that disallow specific commands, you can create one or more role definitions to use those rights. For example, you might create one role definition to run in an unrestricted shell that requires users to invoke dzdo to execute privileged commands and another role definition that runs in a restricted shell but does not require users to execute privileged commands using dzdo. The second role might be useful if you have existing scripts that would have to be modified if invoking dzdo is required.

To create a role definition for specific command rights:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role.

    For example, type a name such as operators and descriptive text such as Users with this role can run privileged commands but not reset passwords, add or delete users and groups.
6. Click **System Rights** if you want this role definition to be used in a restricted shell environment as a replacement for the predefined UNIX Login role.

    To use this role, a user must be assigned to a role definition that has at least one login system right, such as Password login and nonpassword (SSO) login are allowed or Nonpassword (SSO) login is allowed.
7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select all of the command right that disallow specific operations, the command right that grants access to all remaining commands, and a PAM access right, then click **OK**.

For example, you might add the following previously-defined command rights to this role definition:

No password resets
No user adds
No group adds
No user deletes
No group deletes
Root like access (* for all commands not explicitly disallowed)
PAM ssh/login allowed

This role definition allows members of the operators role to execute any command within a restricted shell environment except those explicitly disallowed, including privileged commands, without invoking dzdo first. You can assign the role definition to the appropriate Active Directory users or groups like the previous role definitions.

Create an unrestricted shell role definition that uses the command rights

The command rights were configured to allow execution in either a restricted shell environment or an unrestricted shell environment. In an unrestricted shell environment—for example, the default shell environment when users are assigned the UNIX Login role—commands that require administrative privileges must be executed by first invoking the dzdo command, which is similar to invoking commands with sudo.

You can control whether users are required to enter a password or another form of authentication when they execute privileged commands using dzdo by setting one of the **Re-authenticate** options on the Attributes tab when you create a command right. By default, no password is required. If you were adding a new command right that requires reauthentication, you would click the Attributes tab, then select **Re-authenticate current user** or **Re-authenticate using target user's password**. For more information about these options, see Requiring re-authentication to run commands.

In most cases, the default of no password is appropriate because the user has been previous authenticated before invoking dzdo to execute a privileged command and the **Reauthenticate using target user's password** option requires the user to know the privileged account password. For example, if select this option and the run-as user is root, the user must know the password for the root account.

The steps for creating the role definition that includes the previously-defined command right are the same for the unrestricted shell as for the restricted shell except that, at Step 6 in the topic Create a restricted role definition for the service account, in the System Rights tab you would also select the **Login with non-Restricted Shell** option if you are not using the UNIX Login role. You could add all of the same command rights to the role definition and grant the same privileges and exceptions.

The primary difference between the two role definitions would be how users execute their privileged commands.

In the restricted shell environment, users running the adflush command requiring administrative privileges:

dzsh $ adflush

In the unrestricted shell environment, users running the adflush command requiring administrative privileges:

[tulo@ajax]$ dzdo adflush

Creating a role definition with rescue rights

The Rescue rights option allows you to control which users should be able to log on if problems with authentication, the authorization cache, or the auditing service are preventing all other users from logging on. For example, if you have a computer with sensitive information, such as credit card numbers or intellectual property, you might require auditing for all users in the role with access that computer. If the auditing service is stopped or removed on that computer, no one would be able to log on and use the computer until auditing is restored. If you create a role with the Rescue rights option selected, only the users assigned to that role are able to log on and continue working until the problem that caused the lockout is found and fixed.

Users who are in a role granted access because they have rescue rights can still be audited through the system logging facility. However, their activity is not recorded in the audit store database if the auditing service is not available.

Creating a role definition that allows local users

Most role definitions are only applicable to Active Directory users and groups. In some cases, however, you might want to create a role definition that can be assigned to local users. For example, you might want to assign local users to a role that grants rescue rights to ensure a specific local account can log on if an Active Directory user is not available.

Role definitions that allow local users to be assigned cannot include PAM access rights or SSH rights, however, and therefore do not include any of the UNIX

system rights. You can use role definitions that allow local users to assign specific command rights to local and Active Directory users. You can also set the audit level for the role definitions that allow local users to be assigned.

If you select the option to allow local users, you can specify the local accounts when you assign the role by clicking **Add Local Account**, then typing the name of local UNIX or Windows accounts to assign to the role. The **Add Local Account** option is not displayed when assigning a role definition that does not allow local accounts.

Creating a role definition for secure shell rights

You can add SSH rights to any role definition as long as the role does not accept local users. Although SSH rights require the PAM ssh right, the role definition to which you add SSH rights does not require the PAM access right. As long as a user is assigned to a role that includes the PAM ssh right, you can add SSH rights to any other role definition to make the rights effective.

In addition to adding the rights to a role definition, you must set the ServiceAuthLocation parameter in the sshd_config configuration file to check for secure shell rights when users log on using a secure shell. In most cases, you should use the **Enable application rights** group policy to set this parameter for all Centrify-managed Linux and UNIX computers. This group policy sets the path to the dzsshchk command which verifies the specific applications rights for users when they log on.

Alternatively, you can manually set this parameter on an individual computer by editing the configuration file to include the following:

ServiceAuthLocation /usr/share/centrifydc/libexec/dzsshchk

Creating additional custom roles and role assignments

The previous sections described common role definitions that organizations implement to begin the process of migrating and removing locally defined privileged accounts. For most organizations, locallydefined accounts with privileged access present a security risk and are often identified as a compliance issue by auditors.

By creating role definitions similar to those described in this chapter, you can eliminate the need to share root and service account passwords while still providing privileged access to computers where it's needed. These additional roles are not required, however. You can choose to create them or create a completely different set of role definitions to suit your organization. For example, you might decide to create custom roles specifically tailored to the needs of database administrators, backup operators, and web application developers. Similarly, you might decide to create separate role definitions that are customized with AIX command rights for AIX administrators that are different from the command rights defined for Solaris administrators.

As with the common role definitions, additional custom role definitions can be created in the top-level parent zone and available throughout the zone hierarchy or in any child zone. They can also span all the computers in a zone or be assigned specifically to individual computers.

If you plan to create your own custom role definitions and role assignments, keep the following key points in mind:

- Rights associated with roles are cumulative. Users receive all of the rights in all of the roles they are assigned.
- Users must be assigned at least one role that allows an interactive login or Kerberos authentication to have any access to any computers. For existing users, this is accomplished by assigning the default UNIX Login role during the migration to Active Directory.
- Users must be given the Login with non-Restricted Shell system right to have access to a full shell. If they are in a role without this right, they can only execute the commands explicitly defined for their role.

  For users who have previously had full shell access, this limitation can be frustrating, unexpected, and unworkable. Before placing or moving users into a restricted role, be sure those users and managers throughout the organization are well-informed and wellprepared for the change and understand the business reasons for the change.

Adding custom attributes

You can add custom attributes to role definitions and role assignments. For example, you might want to use a custom attribute to reference a ticket number associated with a specific type of access request, role definition, or temporary role assignment. Custom attributes are optional and you can use them to capture any kind of information that is meaningful to your organization.

You can add custom attributes when defining or modifying a role, defining or modifying a computer role, or when modifying role assignment properties.

Exporting and importing rights and roles

You can export rights and role definitions from any zone if you want to save part or all of the information to a file. You can then import all or part of that information into a new zone and modify it, if needed. For example, you can choose to export all the rights you have defined in one zone but create a completely new set of role definitions for those rights in another zone. Exporting and importing provides a convenient way to copy and paste multiple rights and role definitions at one time.

Rights, roles, and role assignments are all inherited from parent to child zones, so there is no need to import or export any authorization information within a zone hierarchy. However, if you have multiple parent zones—for example, representing different geographical regions—you might want to use export and import to copy authorization information from one geographical region's zone to another.

Exporting authorization information

You can export multiple rights and role definitions to an .xml file that you can then use to import these definitions into another zone. You can also copy and paste or drag and drop individual rights and role definitions between zones.

To export rights and role definitions:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name from which you want to export authorization information.
3. Select the Authorization node, right-click, then click **Export Roles and Rights**.
4. Select the information you want to export, then click **Next**.

   For example, select **All** to export all of the rights and all role definitions. Selecting all or individual role definitions exports all of the rights included in those role definitions.
5. Click **Browse** to specify a location and file name for the export file, then click **Next**.
6. Review the information to be exported, then click **Finish**.

Importing authorization information

You can import multiple rights and role definitions that you have previously exported from a zone and saved to an .xml file. You can also copy and paste or drag and drop individual rights and role definitions between zones.

To import rights and role definitions

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you want to import authorization information.
3. Select the Authorization node, right-click, then click **Import Roles and Rights**.
4. Click **Browse** to navigate to the file that contains the authorization information you want to import, then click **Next**.
5. Select the information you want to import, then click **Next**.
6. Review the information to be imported, then click **Finish**.

Updating rights, roles, and role assignments

When you make changes to rights, roles, or role assignments, these changes take effect on managed computers at the next cache update interval, as set by the adclient.cache.expires parameter or the "Set object expiration" group policy. The default update interval is 10 minutes. If you want changes to take place immediately, you can flush the cache on individual computers.

To flush the cache and update authorization changes:

1. Log on or switch to the root user on the managed computer.
2. Run the adflush command to clear the agent cache. For example:

   /usr/sbin/adflush

Reviewing the fundamentals of role definitions

As discussed in Basic concepts of access rights and roles, rights are fundamental to authorizing user access, you cannot assign rights directly to users. Instead, rights are combined into **role definitions** that reflect the needs of a specific job function, such as database administrator, or the ability to perform a particular task, such as start a web service or run commands that compress or extract files. It is up to you, as an administrator, to decide on the role definitions your organization needs and to assign those custom role definitions to the appropriate users and groups.

Basic access rights require Active Directory users to have a complete UNIX profile and at least one role assignment, for example by using the UNIX Login role, that is in effect in the zone to which a computer is joined. To move beyond basic access rights, you must define custom rights and custom role definitions, then add the specific rights to each role definition.

After you configure a role definition with rights, you can assign it to individual Active Directory users or to Active Directory groups, so that the role applies to all members of the group. By assigning role definitions to groups, you can manage ongoing role-based user access completely through Active Directory.

Working with computer roles

In previous chapters a *role definition* described a specific set of access rights for a user or group, including the period of time when those access rights were in effect. A *computer role* is an association that enables you to make the most of those role definitions. Role definitions grant specific access rights or enforce certain access restrictions. Computer roles enable you to associate role definitions with computers that share a similar function or have a common attribute. This chapter describes how you can configure and use computer roles to manage access rights for different sets of users.

How computer roles provide flexibility

Centrify-managed computers can only be joined to one zone at any time. This limitation makes it difficult to manage granular access rights at the zone level alone. Computer roles enable you to group computers that share a common function or attribute and associate the group of computers with a specific set of role assignments to users or groups. Individual computers can be members of any number of computer roles with different sets of users who have different access rights based on their role assignments.

Computer roles can have multiple role assignments

A computer role associates a group of computers with a set of role assignments. For example, you might have several computers that host Oracle database instances. Using a computer role, you can associate the group of computers that host an Oracle database with one role assignment that grants some users full administrative access. That same computer role can associate the same group of computers with a second role assignment that grants some users access to specific commands that must be run using the oracle account. That same computer role can also associate the same group of computers with a third role assignment that grants application users permission to log on using a secure shell session. As long as the set of computers remains the same, you can use the same computer role to grant different sets of users different access rights.

Managing access using multiple computer roles

Computer roles enable you to manage access rights using multiple filters. For example, you might have several computers that host Oracle database instances. Some of the computers that host an Oracle database might also belong to specific departments, such as the finance or engineering organizations. Some of the computers that host an Oracle database might run Red Hat Enterprise Linux while others have a Solaris operating system. You can use computer roles to grant different sets of access rights based on the criteria you want to use to group the computers. In this example, you might have one computer role for Oracle database servers and their database administrators, another computer role for users in the finance and engineering departments, and another computer role for IT staff who specialize in managing either Linux or Solaris computers.

Computer roles enable you to define access rights using any grouping criteria that makes sense for your organization. In this case, you might have one computer role linked with the Active Directory security group for all Oracle servers, a second computer role linked with the security group that only has computers that belong to the finance or engineering organization, and a third computer role linked with the security group for Linux or Solaris computers. If the set of computers grouped together changes, you should use a new computer role to grant different sets of users different access rights.

Planning to use computer roles

Because computer roles provide you with a great deal of flexibility for defining access rights, you might want to do some planning before you create new computer roles. For example, before you create a computer role you must know the criteria you want to use to group computers into one or more Active Directory security groups. You must also identify the users who will have a common set of access rights based on the computer grouping.

At a high-level, defining a computer role requires the following:

- Identify a unique Active Directory security group for each computer role.

  You should identify an attribute the computers in a particular group share, such as computers in the web farm, that host specific applications, or serve a specific department. You can create the group and add computers to it in Access Manager when you create the computer role, or before creating the computer role using Active Directory Users and Computers.

- Identify the sets of users that share common access rights and create Active Directory groups for them.

  You might want to define multiple sets of user-based roles. For example, a computer role for Oracle servers might require a “database users” group, a “database administrators” group, and a “backup operators” group.

- Identify the access rights and role definitions for each set of user-based roles.

  You might want to create specific rights, role definitions, and role assignments for different sets of users, or use existing roles. For example, the “database users” group might only require the predefined UNIX Login role definition, while the “database administrators” group might require access to privileged commands, and the “backup operators” groups might be only be allowed to run a specific set of commands in a restricted shell.

Creating a new computer role

A computer role is similar to a zone in that it defines a group of computers, a set of users, and specific access rights for a combination of computers and users. However, computer roles do not require a computer to be joined to the zone where the computer role is defined and a computer can be a member of multiple security groups and thus multiple computer roles.

Because computer role assignments define a relationship between a security group of computers, a set of rights in a role definition, and a security group of users, they control who can do what on specific computers. You can change the list of computers or the list of users dynamically simply by changing the security group membership.

What to do before creating a new computer role

Before you create computer roles, you must join a domain and zone. You should also decide on the criteria to use for grouping computers. Each computer might belong to several different security groups to used in different computer roles. Depending on your organization's policies for creating security groups, you might want to prepare one or more Active Directory security groups for Centrify-managed computers.

Rights required for this task

To create computer roles, your user account must be a domain user with the following permissions:

msDS-AzScope This object is listed under a globally unique identifier (GUID) for the Authorization object. For example: CN=cab186af-61a0-4d54-a0dd...	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Read description Read msDS-AzScopeName Read msDS-AzApplicationData Write description Write msDS-AzScopeName Write msDS-AzApplicationData

Who should perform this task

A UNIX zone administrator or a Windows domain administrator who is responsible for adding and maintaining security groups performs this task, depending on your organization's policies.

How often you should perform this task

It is common to create new computer roles any time you identify new criteria for grouping computers and role assignments.

Steps for completing this task

The following instructions illustrate how to create a new computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a new computer role using Access Manager

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name that will contain the new computer role.
3. Expand **Authorization to select Computer Roles**, right-click, then click **Create Computer Role**.
4. Type a name for the computer role and an optional description, then select either **<Create group>** to create a new Active Directory group for computers or **<...>** to search for an existing group of computers to use.

   For example, click **Create group** to create a new Active Directory security group named oracle_servers for the computers that host Oracle database instances. If creating a new group, you are prompted for the location, group name, and scope.
5. After you have selected or created an Active Directory security group, click **OK**, then click **OK** to save the new computer role.

   Note: If you're using classic zones, you cannot add cross-forest groups to roles at this time. All groups added to roles should be defined in the local forest. However, users from a trusted forest may be added to groups in the local forest and then added to a role, or they may be directly added to a role. (Ref: IN-90001)

Delinea

Adding computers to a computer role

After you create an Active Directory security group for computers and associate it with a computer role, you can add or remove computers simply by updating the group membership. For example, if you have a computer role for managing access to Oracle database servers and you deploy a new instance, you simply add the new server to the computer security group you created for Oracle servers. You can update the group membership using Active Directory Users and Computers, Access Manager, ADEdit, or another tool of your choice.

After you have specified the Active Directory security group you want associated with a computer role, the account membership is synchronized so you can use Access Manager or another program to make changes.

Steps for completing this task

The following instructions illustrate how to add computers to a computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To add computers to the computer role using Access Manager

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer role to which you want to add computers.
3. Expand Authorization and Computer Roles, then expand the computer role to which you want to add computers.
4. Select Members, right-click, then select **Add Computer**.
5. Type all or part of a computer name, then click **Find Now** to search for the computer accounts to add.
6. Select one or more computers from the results, then click **OK** to automatically add computers to the Active Directory group associated with the computer role.

Adding role assignments to a computer role

For computer roles to be effective, you must create the access rights and role definitions for different sets of users. You can then assign the appropriate predefined or custom roles to different sets of users to grant or restrict their rights within the scope of the computer role. With proper role definitions and role assignments, you can manage access rights for computers completely through group membership. For example, after you have created the role definition for Oracle database administrators, you can add and remove group members to the group you created for Oracle administrators in Active Directory.

For information about creating access rights and role definitions, see the following:

- Defining rights to run privileged commands
- Defining a restricted shell command right
- Adding specific PAM access rights
- Combining secure shell rights
- Creating and assigning custom role definitions

After you have create the appropriate access rights and role definitions, you must assign those roles to the appropriate users and groups to complete the configuration of the computer role.

Steps for completing this task

The following instructions illustrate how to add role assignments to a computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To associate user role assignments with a computer role using Access Manager

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer role to which you want to add role assignments.
3. Expand Authorization and Computer Roles, then expand the computer role to which you want to add role assignments.
4. Select Role Assignments, right-click, then select **Assign Role**.
5. Select the role definition that you want to add to the computer role, then click **OK**.
6. Click **Add AD Account** to search for and select an Active Directory user or security group to assign to the role.

   You can select User or Group as the object to find, type all or part of the user or group name, then click **Find Now**. For example, type "ora" to search for and select the "oracle_db_admins" Active Directory group then click **OK**.
7. Click **OK** to complete the role assignment for the selected user or group in the selected computer role.

Repeat these steps for each role definition you want to assign to users and groups in this computer role. For example, if you have an Active Directory "oracle_db_users" group that should be allowed to log on and run shell commands on the computers in the "oracle_servers" computer role, you would select the predefined UNIX Login role in Step 5 and assign that role definition for the computers in the "oracle_servers" computer role to the "oracle_db_users" group in Step 6.

Viewing and modifying a computer role

You can view information about computer roles by expanding Authorization and Computer Roles for a zone. However, computer roles are also closely linked to the Active Directory groups that define their scope and role assignments, so there are several different ways you might view or modify information about a computer role. For example, you might use Access Manager, Active Directory Users and Computers, or ADEdit commands, depending on what you are trying to do.

In Access Manager, you can expand a computer role, then select **Role Assignments** to see the users, groups, and role definitions that have been assigned on the computers that are members of the computer role. You can also expand a computer role, then select **Members** to see the computers to which the role assignments apply. To see the Active Directory group assigned to the computer role in Access Manager, select the computer role, right-click, then select **Properties**.

If you are using Active Directory Users and Computers, you can view the properties for the Active Directory group associated with the computer role and click the **Members** tab to see the computers assigned to the computer role.

If you want to add a computer to an existing computer role, you can simply add that computer to the Active Directory group associated with the computer role without making any changes in Access Manager. Similarly, if users join or leave your organization, you can simply add or remove those user accounts in the appropriate Active Directory groups that are associated with the computer role. For example, if you define the oracle_servers computer role to associate a specific set of computers with a role assignment that grants administrative rights to users in the Active Directory security group oracle_db_admins, you could simply add the user account for Frank.Smith to the Active Directory security group oracle_db_admins to give that user administrative access on the computers that are members of the oracle_servers computer role. You do not need to make any changes in Access Manager.

To modify the rights and role assignments for a computer role, you must use Access Manager or ADEdit commands.

Using computer roles

Deciding how best to use computer roles requires some planning and configuration that might not be part of your initial deployment plan. To make effective use of computer roles, you must also prepare appropriate role definitions for different sets of users. However, computer roles provide a powerful and flexible option for managing access to computers using your existing processes and procedures for managing Active Directory group membership.

After you create a computer role, it is easy to manage even as your organization changes and grows. For example, if another Oracle database server comes online, you add it to the computer group you created for Oracle database servers in Active Directory. If other DBAs join your organization, you add them to the Active Directory group you created for Oracle administrators. The computer role links the computer group to the role assignment and no additional updates are needed to accommodate these kinds of organizational changes. If you need to modify the access rights, you can change the role definition and have the changes apply to all members of the group.

Requiring multi-factor authentication using computer roles

Computer roles enable you to group and provide access to computers through role assignments. One strategy you might find useful is to use computer roles to control where multi-factor authentication should apply. For example, you might have several computers with highly sensitive material where you want to ensure all user access will require multifactor authentication. To accomplish this goal, you can configure a computer role, then add and remove computers with sensitive information to control whether multi-factor authentication is required.

To require multi-factor authentication based on a computer role

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new computer role.
3. Expand Authorization to select Computer Roles, right-click, then click **Create Computer Role**.
4. Type the role name and, optionally, a role description, then select **<Create group>** for the Computer group to create a new Active Directory group for computers.

   For example, to create a new Active Directory security group for the computers with sensitive information, click **Browse** to select the Active Directory location for the new group. If you are using the default deployment structure, you would browse to a location similar to acme.pubs.org/Acme/Computer Roles then type a group name such as mfa_required_servers, select a scope, and click **OK**.
5. Click **OK** to save the new computer role.
6. Add the computers that require multi-factor authentication for access to the mfa_required_servers Active Directory security group.

   As you add computers to the Active Directory security group, the computers are listed as Members of the computer role.
7. Expand the computer role you creates in Step 4, select Role Assignments, right-click, then select **Assign Role**.

   For example, if you created a new computer role with the role name CR_MFA_required, expand that computer role name to select Role Assignments, right-click, then select **Assign Role**.
8. Select the predefined require MFA for login role definition, then click **OK**.
9. Select **All Active Directory accounts**, then click **OK**.

Working with managed computers

This chapter describes how to add Linux and UNIX computers to Active Directory domains, manage computer accounts and properties, perform common administrative tasks, and leave the domain.

Identifying who can add computers to the domain

Who can join computers to a domain depends on your organization's policies and those policies are enforced through Active Directory. In general, there are two common scenarios:

- Any authenticated domain user can add up to ten computers to the domain.

  This is the default behavior for Windows computers. Many organizations follow this policy, so that administrative access is not required to add computers to a domain.

- Only users with specific permissions can add computers to the domain.

  Some organization restrict who can add computers to the domain. For example, a user might have to be a member of the Domain Admins or Account Operators group to add computers to a domain.

The policy your organization follows for Windows also applies when you want to add Linux and UNIX computers to a domain. If any authenticated user can add a Windows computer to the domain, adding a Linux or UNIX computer does not require an administrative user name and password. If only administrative or delegated users are allowed to add computers to the domain, the user adding a Linux or UNIX computer must provide an administrative or delegated user name and password.

If you aren't sure whether an administrative account is required to join a domain, you can prepare computer account before attempting to join the domain, and allow the computer account itself to be used to join the domain. Performing this type of "selfservice" join simplifies the operation and allows the computer account to manage its own password without administrative intervention.

Preparing computer accounts before joining

If joining the domain is restricted to privileged users, or if you know that you will need to specify computerlevel overrides, you can prepare computer accounts in advance for the Linux and UNIX computers you want to add to the domain.

There are several advantages to preparing computer accounts before joining the domain. For example, preparing a computer account enables you to accomplish the following:

- Specify the user, group, or computer account with permission to join the computer to the domain.
- Define the organizational structure you want to use for computers in Active Directory.
- Delegate administrative tasks for managing the computer account.
- Specify the user or group with permission to manage computer-level overrides for the computer.

By preparing the computer account in advance, you can minimize the changes or configuration steps you might otherwise have to perform after joining the domain. For example, by identifying the account to use when a computer joins the domain you can ensure users can add their own workstations without being assigned any special rights. By selecting the appropriate organizational unit for the computer account ahead of time, you minimize the need to move the computer account after joining the domain.

To prepare a computer account using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the computer account.
3. Right-click, then click **Prepare UNIX Computer**.
4. Select the type of preparation you want to perform, then click **Next**.

   In most cases, you should select both options to ensure the appropriate user or group has the permissions required to join the domain and set computer-level overrides.
5. Choose whether to create a new computer object or select an existing computer object, then click **Next**.

   If the computer account exists, but you want to add a zone profile and delegate permission to join the domain and manage computer overrides for the computer, click **Browse** to search for and select the existing computer object. After selecting an existing computer account, click **Next** to continue to Step 7.
6. Type the computer name to use for the new computer account and specify a location for the computer account object in Active Directory, then click **Next**.
   - For **Computer name**, type the host name to use for the computer account in Active Directory.
   - For **Domain**. verify the domain name displayed is the appropriate domain for the computer account to join. Click **Browse** to navigate to a different Active Directory domain.
   - For **DNS name**, verify the DNS name for the computer account. You can modify the DNS name for the computer, if needed. For example, if computer names in DNS use a different suffix than the Active Directory domain, you might need to modify the default value displayed.
   - Select **Create the computer object in the container** to specify the parent container for the new computer account in Active Directory. In most cases, you should use the default parent container object. Click **Change** to navigate to a different container object for the computer account.
7. Select the **Allow this computer to join the domain using a read-only domain controller** option if you want the computer to join itself to the domain using a read-only domain controller and select the type of license to use, then click **Next**.

   If you click **Next** without selecting **Allow this computer to join the domain using a read-only domain controller**, the computer must join the domain by connecting to a writable domain controller.
8. Review the default list of service types and service principal names for the specified computer, then click **Next** to accept the default set of service principal names.

If you want to make changes to the default services or service principal names, you can do the following:

- Click **Add** to add a service type or add a new service name to an existing service type.
- Select a service principal name and click **Edit** to change the name.
- Select a service principal name and click **Remove** to delete the name.
- Click **Default SPN** to restore the default list of service principal names.

  If you are in an environment where multiple instances of the same SPN are possible, as a user with administrator privileges, use the -d or --forceDeleteObjWithDupSpn parameter with the adjoin command to ensure duplicate SPNs are removed.

9. Select whether to allow a specific user or group to join the computer to the domain or use the computer account and automatically-generate password to join the domain, then click **Next**.

   In most cases, select **Allow the computer to join itself to the zone** to allow the computer account to perform a "self-service" join. This option is selected by default because it allows you to automate the join operation so that a user name and password are not required.

   If you want a specific user, group, or computer account to be used to join the domain, select **Allow this user, group, or computer to join the computer to the zone** then click **Browse** to search for the user, group, or computer that you want to give permission to join the computer to the domain.

10. Select the user, group, or computer account with permission to set computer-level overrides, then click **Next.**

    By default, the permissions required to manage computer-level overrides are granted to members of the Domain Admins group. You can click **Browse** to search for and select another user, group, or computer account.

11. You can choose to skip permission delegation, if desired.

    If you select this option, the service does not set the security descriptor for the computer; you'll need to go in and set that attribute yourself. Some organizations prefer to set security descriptors manually. Security descriptors include security information such as the object owner, who has access rights to the object, and so forth.

12. Review your configuration settings, then click **Next**.

13. Review the confirmation of the operation performed, then click **Finish**.

    The computer account is created in Active Directory and a zone profile for the computer is added to Access Manager in the zone's Computers container. The user or group you have designated as the trustee can now join this computer to the domain using the adjoin selfserve command line option, and the group you designated for computerlevel overrides can add users and role assignments to the computer.

Delegating permissions when preparing a computer account

When you prepare a computer account, you have the option to grant a specific user, group, or computer account the administrative permissions required to perform two separate tasks:

- The permissions required to join the computer account to the domain.
- The permissions required to set and manage computer-level overrides

In most cases, you should select both options even if you want to grant different accounts the permissions required to perform each task.

However, it is possible to create a computer account and not delegate permission for computer-level overrides by deselecting the **Delegate permission for machine overrides** option. If you deselect this option, you are the only administrator who can set profile or role assignment overrides for the computer. No other user or group will be granted the permissions required to set or manage computer-level override for user profiles or role assignments.

Likewise, it is possible to delegate permissions for computer-level overrides without preparing the computer to join the domain by deselecting the **Prepare computer for adjoin** option. If you deselect this option, the computer icon appears in the zone, but the Active Directory computer object and service connection point are not created. The designated trustee can set computer-level override for user profiles or role assignments. No other user, group, or computer account will be specifically granted the permissions required to join the domain.

If any authenticated user can add computers to the domain, then any user with a valid domain account can join Linux and UNIX computers to the domain. If adding computers to a domain requires an administrative account, only the administrator who creates the computer account can join it to Active Directory.

For more information about who can add computers to a domain, see Identifying who can add computers to the domain.

Allowing password resets for computer accounts

If you use Access Manager and the Prepare UNIX Computer wizard to create a computer account before joining the domain, you can select the **Allow the computer to join itself to the zone** option to set the permissions required for a computer to manage its own account. If you use Active Directory Users and Computers to create a computer account, however, you need to manually modify the permissions for the account.

By default, most computer accounts do not have permission to reset their own account password. This prevents the delegation of administrative rights for the computer to the local computer account. If you want to give a computer account administrative rights in a zone, you need to modify the computer account to allow password resets. In addition, allowing a computer account to update its own properties enables Access Manager to display the agent version and maintain operating system information for the computer account.

Checking for the appropriate permissions

To check whether a computer account allows password resets, you can view the permission settings for the account.

To check and modify the permissions for a computer account:

1. Open Active Directory Users and Computers, expand the domain, and select Computers to find the computer account to which you want to assign administrative rights.
2. Select the computer account, right click, then select **AD Properties**.
3. Click the **Security** tab, scroll down the list of group or user names and select **SELF**.
4. In the list of Permissions for SELF, scroll to the **Reset Password** permission, click **Allow**, then click **OK**.
5. Select the computer account, right-click and select **Reset Account**, then click **Yes**. When the account is reset, click **OK**.

Assigning administrative rights to computer accounts

After you have checked the Active Directory permissions for a managed computer account and modified them, if necessary, you can assign zone administrative rights to the account through Access Manager.

To give administrative rights to the computer account:

1. Open the Access Manager console.
2. In the console tree, select **Zones**, and if necessary, **Child Zones**, then select and expand the zone in which you are interested.
3. Right-click, then click **Delegate Zone Control**.
4. Click **Add**, select **Computer** from the Find list, then click **Find Now**.
5. In the results, select **Domain Computers**, click **OK**, then click **Next**.
6. Click **Join computers to the zone** and optionally, **Remove computers from the zone**, then click **Next**.

   Note: {/b}In most cases, these are the only administrative tasks you should assign to the computer account. You can, however, give the account additional rights, if needed. For information about the permissions associated with each delegated task, see the *Planning and Deployment Guide*.
7. Click **Finish**.

Joining a domain

To begin authenticating users and authorizing access to Linux and UNIX computers and resources, you must first add the computers you want to manage to the appropriate Active Directory domains in one or more Active Directory forests. You can do this by running the adjoin command interactively or by using the adjoin command in a script. A successful join operation is what converts a Linux or UNIX computer into a **Centrify-managed computer**.

Connecting to the domain controller

To add a new computer to a domain, you must specify the domain you want to join. The adjoin program then locates an appropriate domain controller for the domain you specify and connects to Active Directory through that domain controller. By default, the domain controller to contact is determined by the Active Directory site topology. If the nearest domain controller in the site is not available, the agent attempts to connect to the next closest domain controller in the site. If no domain controller can be contacted or the connection takes too long to complete, the join operation fails.

If you don't want to agent to select a domain controller based on the site topology, you can specify a master domain controller on a zone-by-zone basis. If you specify a master domain controller, the agent will connect to the appropriate domain controller based on the zone you are joining.

What happens during the join operation

If the Centrify Agent can successfully connect to an Active Directory domain controller, it performs a series of key tasks to complete the join operation. For example, during the join operation, the adjoin program completes the following tasks:

- Starts the Centrify Agent for *NIX adclient process.
- Checks whether a computer account already exists for the local computer in Active Directory. It creates a new Active Directory computer account for the local computer, if needed.
- Sets the password on the Active Directory computer account to a randomly-generated password. The password is encrypted and stored locally on the UNIX host to ensure that only the Centrify Agent has control of the account.
- Updates the Kerberos service principal names used by the host computer, generating new a Kerberos configuration file and krb5.keytab entries, and generating new service keys for the host and http services.
- Synchronizes the local computer's time with Active Directory to ensure the timestamps for Kerberos tickets are accepted for authentication.

After joining a domain

By default, computers function exactly the same after joining the domain as they did before joining the domain. Local users can continue to log on and existing programs and applications can continue to work as they did before joining the domain. The primary difference after joining the domain is that you have more complete control over access to the computer and what Active Directory users who are granted access can do. You will also have more tools at your disposal for managing computer properties and operations. For example, after joining a domain, you can use any combination of the following tools:

- Access Manager
- Access Module for Windows PowerShell
- ADEdit command line programs and scripts
- Active Directory Users and Computers
- Group Policy Management console and Centrify group policies

You can use any of these tools to add Active Directory users to the appropriate zones, and to define and assign appropriate rights and roles for the users who need access to Linux and UNIX computers.

Joining a domain and zone with the adjoin command

In most cases, you add a computer to the domain by running the adjoin command directly on a local computer. You run this command once for each Linux or UNIX computer you want to add to a domain in the forest. Using the administrator or a designated user account, you can run the command interactively at the command line or include the command in a script to automate joining a domain.

Specifying the most common arguments

Whether you join the domain interactively from the command line or using a script, you must specify a few required arguments. You might also need to specify several additional arguments, such as a user name and password for an account with permission to join the domain, an alias for the computer in Active Directory, or the organizational unit in which to place the computer.

The most common format for the adjoin command is:

adjoin --user username --zone zonename domain

For example, the following command illustrates the most common format for the adjoin command:

adjoin --user shea@acme.com --zone LinuxDev sales.acme.com

This command connects to Active Directory as the user shea@acme.com to add the local computer to a previously-created zone called LinuxDev zone and to the sales.acme.com domain. In this example, the zone and domain name are required. The user name is not a required argument—if not specified the adjoin command would prompt for the Administrator account password. However, because the user shea is a member of the acme.com domain rather than the sales.acme.com domain, the user account must be specified in the user_name@domain_name format.

Because the password is not specified in the command line, the adjoin program prompts for the Active Directory password to authenticate the shea@acme.com account before connecting to Active Directory.

In most cases, you should avoid including the password for an account as part of the adjoin command line for security reasons. If you are using adjoin in a script, however, you must include the --password option or provide another mechanism for inputting a valid password. For more information about adjoin command line options and running adjoin commands, see the adjoin man page.

If the adclient process is able to connect to Active Directory and the join is successful, a confirmation message is displayed. By default, the join operation adds the new computer account to Active Directory in the domain_name/Computers container. If the connection to Active Directory fails, a warning message is displayed and the join operation fails.

Using the self-serve option for a previously-created computer account

If you have previously prepared a computer account in Active Directory as described in Preparing computer accounts before joining, you can use the selfserve (-S) option to join a domain without specifying a user name and password. For example, you can run a command similar to the following to join the domain:

adjoin --selfserve domain

For example:

adjoin --selfserve cendura.org

Note that you must specify the domain to join but not the zone—the computer is automatically joined to the zone in which the computer object was pre-created.

If you want to preserve service principal names (SPN) configured in the centrifydc.conf, use the adjoin command option -r spn or --useConf spn. This option only works in conjunction with the -S, --selfserve command.

Joining a domain in workstation mode

In most cases, zones are required if you are adding Linux and UNIX computers to Active Directory to address account migration and role-based access rights. However, it is possible to deploy without using zones to organize computers, rights, roles, and other information.

The workstation mode is intended for computers that function in the same way as Windows workstations where any valid user can log on to any computer that is joined to the domain. In general, workstations do not require you to manage identity attributes, such as UIDs and GIDs, or access-related attributes, such as the hours a user is allowed to log on. To mirror this behavior for Linux and UNIX computers, the workstation mode automatically creates a local user profile for users when they log on and does not apply any access rules unless you configured them for the user account in Active Directory.

Computers that join the domain using workstation mode are added to a single Auto Zone and are treated the same as Windows workstations, and are managed by Active Directory and group policy settings. You can use the workstation mode and Auto Zone for any computers that do not require profile management or role-based access controls. You can also have any combination of workstation computers that don't require profile management and access control and workstations and servers that do require profile management, access control, hierarchical zones. For more information, see Using workstation mode and Auto Zone.

To join a domain using workstation mode instead of zones, you can run a command similar to the following:

adjoin --workstation --user username domain

For example:

adjoin --workstation --user kai.rodriguez cendura.org

This command adds the local computer to a single Auto Zone. The Auto Zone requires no configuration and there are no properties, user profiles, or access rights to manage. All Active Directory users and groups in the forest, or in forests with a two-way trust, can access the computers in the Auto Zone.

Joining the domain using the computer account

On the computer to which you have given administrative rights, run the adjoin command and set the user name parameter to the computer name with a dollar sign ($) appended and the password to the computer name.

adjoin domain --zone zoneName --user computername$ --password computername

For example, if the computer name is valencia and the Active Directory domain is arcade.com, you would run a command similar to the following:

adjoin arcade.com --zone finance --user valencia$ --password valencia

Setting the password interval for managed computers

After joining a domain, the password for the managed computer account in Active Directory is automatically reset at a regular interval to ensure security. By default, the password for the computer account is updated with a new, randomlygenerated password every seven days. You can customize how frequently the password for the account is changed through the **Password change interval** group policy or by modifying the adclient.krb5.password.change.interval parameter in the configuration file, centrifydc.conf, on any managed computer.

Allowing a managed computer to authenticate NIS users

If you are using one or more managed computers as a NIS server to provide "agentless" authentication to NIS client requests or to publish NIS network maps, you can identify those computers in Access Manager by setting the **Allow this computer to authenticate NIS users** option on the computer's Centrify Profile. Setting this option adds the computer account to the zone_nis_servers Active Directory group. Additional configuration is required if you want a managed computer to respond to NIS client request.

This option is provided because using Centrify Network Information Service (adnisd) is more secure than using a legacy NIS server and can be useful to accommodate certain situations, such as a transition from NIS domains to Active Directory domains. However, continuing to use NIS client requests to retrieve network information is not a secure practice and might result in an audit finding in some organizations.

For more information about installing, configuring, and using the Centrify Network Information Service, see the *Network Information Service Administrator's Guide*.

Changing the zone for a managed computer

When you join a domain, you must join a specific zone unless you are using workstation mode and connecting through Auto Zone. Over time, you might want to migrate managed computer accounts from one zone to another. You can change the zone information for a computer at any time, if needed. You can change the zone for a computer by selecting a new parent or child zone in the computer properties or by cutting and pasting the computer object from one zone to another in Access Manager.

After you change the zone for a managed computer, you must restart the Centrify Agent for *NIX on that computer for the change to take effect. You are not required to run adleave or adjoin to complete the change. For example, after you have changed the zone for a computer log on to that computer and run the following command:

/etc/init.d/centrifydc restart

Alternatively, you can restart the managed computer to restart all services, including the Centrify Agent for *NIX. In most cases, however, restarting the agent is sufficient.

Note: If the computer has role assignments defined, you might be prevented from moving the computer until you remove the role assignments.

Changing domain information for a managed computer

Once a computer successfully joins a domain, you can remove it from a domain at any time by using the adleave command. You must also use the adleave command before you can join a new domain or make changes to the domain information for a computer, such as changing the computer name.

Leaving a domain

Leaving the domain before attempting to join a new domain or changing a computer name ensures that there will not be file conflicts or orphaned information that might prevent the join operation from completing.

You should note that leaving the domain removes all of the Centrify-specific information for the managed computer from Active Directory and reverts any computer settings that were changed by the adjoin command to their preadjoin condition. These changes include reverting PAM, NSS, and Kerberos configuration files to their pre-adjoin states and deleting the /etc/krb5.keytab file. Leaving the domain does not delete the Active Directory computer object itself.

Leaving the domain does not delete the Active Directory computer object itself. If you want to completely remove any record of the computer from Active Directory, you must delete the computer object using Active Directory Users and Computers.

Joining a different domain

After running the adleave command, re-run the adjoin command with the appropriate arguments to join a different Active Directory domain. For example:

adjoin --zone arcade.com --user gale.harris operations.acme.com

For more information about using the adjoin and adleave commands, see the adjoin or adleave man page.

Renaming a managed computer

If you need to rename a Linux or UNIX computer that is joined to a domain, you should first leave the domain, rename the computer, then rejoin the domain. Otherwise, you could have issues with the service connection point or service principal name for the computer.

Customizing configuration settings for a computer

You can configure many aspects of the environment for individual computers by applying a Group Policy Object to a site, domain, or organizational unit that includes managed computers and enabling Centrify-specific group policies. For example, you can use policies to customize PAM operations, the length of time to wait for connections between the Centrify Agent for *NIX and Active Directory, or how frequently to change the computer account password. For information about the group policies available and how to enable them, see the *Group Policy Guide*.

If you are not deploying Centrify group policies, you can also customize the configuration settings in any computer's local agent configuration file, centrifydc.conf. The comments within the file describe the most common settings. For more information about setting the parameters directly in the agent configuration file, see the *Configuration and Tuning Reference Guide*.

Enabling FIPS-compliant encryption

The Federal Information Processing Standard 140-2 (FIPS 140-2) describes US Federal government requirements that IT products should meet for sensitive, but unclassified use. The standard is published by the National Institute of Standards and Technology (NIST) and is required by all non-military agencies of the United States Government. This standard is also widely used by many other organizations outside of the government.

The standard defines the security requirements that must be satisfied by a cryptographic module used to secure unclassified information. There are four levels of security: from Level 1 (lowest) to Level 4 (highest). These levels are intended to cover the wide range of potential applications and environments in which cryptographic modules might be deployed. The security requirements cover areas related to the secure design and implementation of a cryptographic module.

The Centrify Agent can be configured to use FIPS-compliant encryption so that a managed computer can successfully join a domain that is FIPS 140-2, Level 1, compliant.

Verifying the Windows environment

Before you configure the Centrify Agent to use FIPS-compliant encryption, you should verify that the Active Directory domain meets the minimum requirements for FIPScompliance. For a Centrify-managed computer to join a FIPS 140-2 Active Directory domain, the Active Directory domain must meet the following basic requirements:

- The domain must be at domain functional level Windows Server 2008, or later.
- The forest must have a global catalog computer that is running at domain functional level Windows Server 2008, or later.
- The domain must have at least one Windows Server 2008 R2, or later, domain controller.
- Any trusted domains you plan to access must be at domain functional level Windows Server 2008, or later.

Although a managed computer can successfully join a domain that has trust relationships to domains at a lower functional level, it cannot access users in those trusted domains, for example, to add user profiles or roles to a zone.

Using group policy for FIPS compliance

If your Active Directory forest meets the minimum requirements and you have configured the Windows environment with the local or group "System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing" security policy, you can make Centrifymanaged computers FIPS-compliant by enabling and applying the Centrify "Use FIPS compliant algorithms for encryption, hashing and signing" group policy. You should not use the equivalent Windows group policy to configure FIPScompliant communications for Linux and UNIX computers. The Centrify "Use FIPS compliant algorithms for encryption, hashing and signing" group policy is specifically designed to support Active Directory domains that are configured for FIPS 140-2 compliance.

The Centrify "Use FIPS compliant algorithms for encryption, hashing and signing" group policy is defined in a separate XML (centrifydc_fips.xml) or ADM (centrifydc_fips.adm) template file. The template file is included in the Centrify group policy extension. You must add one of these templates to a Group Policy Object to make a Centrify-managed computer FIPS-compliant mode. For information about adding template files and enabling group policies, see the Group Policy Guide. After you enable the policy, it takes effect at the next group policy update interval. To have the policy applied immediately, run the adgpdupdate command.

Using the XML template group policy

If you use the XML group policy template to enable FIPS mode, the policy verifies that each computer is joined to a domain at the domain functional level Windows Server 2008, or later. If a domain controller does *not* meet this minimum domain functional level, the policy issues a warning that allows you to skip enabling of FIPS mode for that computer.

The XML group policy template also verifies *all* computers to which the policy applies are running a supported operating system. On the computers that are running a supported operating system, the policy sets the fips.mode.enable configuration parameter to true and automatically stops and restarts the adclient process. After the restart, the computers where the policy was applied are FIPS-compliant.

If the computer is *not* running a supported platform, the XML policy leaves the fips.mode.enable configuration parameter set to false, and does not stop and restart adclient. The computer remains joined and the current encryption and hashing algorithms remain in force.

Modifying the agent configuration file

The Centrify "Use FIPS compliant algorithms for encryption, hashing and signing" group policy sets the fips.mode.enable parameter in the Centrify

configuration file to true. By default, this parameter is set to false until the group policy is applied and the computer is updated at the next group policy update interval. You can also manually modify this parameter setting directly in the agent configuration file (centrifydc.conf), then restart the adclient process to enable FIPS mode. In most cases, however, you should use the group policy to set the configuration parameter to enable FIPS mode rather than manually editing the fips.mode.enable parameter on individual computers.

Applying the group policy to a domain

In most cases, you should apply the "Use FIPS compliant algorithms for encryption, hashing and signing" group policy to a Windows Server 2008, or later, domain to enable FIPS mode. If the group policy is applied to the domain, then the computer will be enabled for FIPS mode automatically when it joins the domain.

Agent requirements for FIPS-compliant encryption

You can only configure FIPS mode for Centrify Agents, version 5.0.2, or later. In addition, FIPS mode is only supported on specific distributions of Linux and Mac OS X operating systems. For a complete and up-to-date list of the platforms that Centrify supports in FIPS mode, see the NIST validation entry for Centrify FIPS mode.

NTLM authentication

The Centrify Agent does not support NTLM authentication through SMB or SMB2 when configured to use FIPS-compliant encryption. FIPS mode only allows NTLM pass-through authentication over SChannel. Note that NTLM pass-through authentication requires a Windows Server 2008 R2, or later, domain controller.

Non-compliant operations

When configured to run in FIPS mode, the agent uses non-FIPS compliant *hash* and *key-hash* algorithms, as follows:

- MD4, MD5 and HMAC-MD5 are used to support NTLM passthrough authentication (including using NLTM for PAM authentication).
- MD4 is used to generate the managed computer password hash for use in setting up AES NetLogon Secure Channel. AES NetLogon Secure Channel is used for NTLM pass-through authentication as well as for updating operating system version attributes.
- MD5 is used to generate the UNIX password hash to verify against the MD5 password hash that is stored in the cache during disconnected mode login. (This is for backward compatibility support; this happens when you upgrade from a DirectControl version that does not support the SHA256 password hash.)

When configured to run in FIPS mode, the agent uses a non-FIPS compliant *encryption* algorithm, as follows:

- Non-FIPS compliant encryption will be used in encrypting secret information for internal communication through a UNIX domain socket.
- A non-FIPS compliant random number generator is used in generating the Initialization Vector used in the encryption.

Configuring the encryption types for trusted domains

Inter-realm keys for the AES256-CTS and AES128-CTS encryption types must be established between any trusted domains to enable Active Directory users from these domains to log on to the joined computer. You can use the ksetup utility, installed by default on the domain controller, to set up the inter-realm keys.

To configure the inter-realm keys

1. On the domain controller, open a Command Prompt window.
2. Type the following commands:

   C:\>ksetup.exe /SetEncTypeAttr trustedDomain AES256-CTS-HMAC-SHA1-96

   C:\>ksetup.exe /SetEncTypeAttr trustedDomain AES128-CTS-HMAC-SHA1-96

   Note: If you are using pre-validated Active Directory users, you must enable these users for Kerberos AES 128- and 256-bit encryption. You can do so by editing user accounts in Active Directory or by setting attributes for the users in ADSI Edit. For more information, see Enabling required encryption types for pre-validated users.

Manually granting write permissions for a computer account

If the domain that the managed computer is joining does not have at least one Windows Server 2008 R2 domain controller, you must manually grant write permission for the Operating System Version and msDS-supportedEncryptionTypes attributes to the computer account of the joined computer.

To grant write permission for required attributes to the computer account

1. Open Active Directory Users and Computers or ADSI Edit.
2. Expand the Computers container and select the computer that is joining the domain, right-click, then click **Properties**.
3. Click the Security tab, then click **Advanced**.
4. Click **Add**.
5. In the "Enter the object name to select" field, type SELF and click **OK**.
6. Click the Properties tab, select **This object only** from the Apply to list, then scroll down and click **Allow** for the following attributes:
    - Write msDS-supportedEncryptionTypes
    - Write Operating System Version
7. Click **OK** in each dialog box to close the dialog and save the new permissions.

Manually granting write permissions for a user account

If the domain that the managed computer is joining does not have at least one Windows Server 2008 R2 domain controller, you must manually grant write permission for the Operating System Version and msDS-supportedEncryptionTypes attributes to the user account used to join the computer to the domain.

1. Open Active Directory Users Computers or ADSI Edit.
2. Expand the Computers container and select the computer that is joining the domain, right-click, then click **Properties**.
3. Click the Security tab, then click **Advanced**.
4. Click **Add**.
5. In the "Enter the object name to select" field, type the name of the Active Directory user who will join the computer to the domain and click **OK**.
6. Click the Properties tab, select **This object only** from the Apply to list, then scroll down and click **Allow** for the following attributes:
    - Write msDS-supportedEncryptionType
    - Write Operating System Version attributes
7. Click **OK** in each dialog box to close the dialog and save the new permissions.

Enabling required encryption types for pre-validated users

If you are using pre-validated Active Directory users, you must enable Kerberos AES 128- and 256-bit encryption for these users. You can do so by editing the user accounts in Active Directory Users and Computers or by setting attributes for the users in ADSI Edit.

To enable encryption for pre-validated users by using Active Directory Users and Computers

1. On the domain controller, open Active Directory Users and Computers.
2. Navigate to the domain and select **Users**.
3. Select the pre-validated user, right-click, then click **Properties**.
4. Click the Account tab, then select the following Account options:
    - This account supports Kerberos AES 128 bit encryption.
    - This account supports Kerberos AES 256 bit encryption.

5. Click **OK** to save the updated account information.

To enable encryption for pre-validated users by using ADSI Edit

1. On the domain controller, open ADSI Edit.
2. Navigate to the domain and select **CN=Users**.
3. Select the user, right-click, then click **Properties**.
4. In the Attribute Editor tab, select the msDS-supportedEncryptionTypes attribute and select **Edit**.
5. Type 0X18 to set the hex value for the attribute and click **OK**.

   You should see that the value shows:

   0x18 = (AES128-CTS-HMAC-SHA1-96 | AES256-CTS-HMAC-SHA1-96)
6. Click **OK** to save the new setting.

How Centrify FIPS mode affects other encryption settings

If you enable FIPS mode, you cannot specify the Data Encryption Standard when joining the domain. The adjoin --des option is not supported. Only AES authentication is supported.

If you have specified multiple types of encryption for the computer by setting the adclient.krb5.permitted.encryption.types parameter in the centrifydc.conf configuration file, only aes256-cts and aes128-cts encryption type keys are generated and saved to the keytab file. However, if arcfour-hmac-md5 encryption is specified, the MD4Hash of the computer password is generated and saved to the keytab file.

In addition, depending on how your environment is configured, you can choose whether to remove any non-AES encryption keys for service principal names (SPNs) from the computer's keytab file by setting the adclient.krb5.clean.nonfips.enctypes parameter in the centrifydc.conf configuration file. If you set this parameter to true, adclient scans the keytab file and removes any non-AES encryption keys for SPNs during startup. This parameter is false by default.

Restarting the agent after enabling FIPS mode

If you use the ADM group policy template, which does not perform validation checks, or if you manually enable FIPS mode by setting the fips.mode.enable parameter in the agent configuration file, the adclient process will not start if the domain functional level is below Windows Server 2008.

If you attempt to start adclient and the domain functional level is below Windows Server 2008, you will see the following error message:

Cannot start adclient in FIPS Mode as machine is joined to domain with PreWindows 2008 Domain Functional Level!

To restart the agent, you must disable FIPS mode by setting the fips.mode.enable parameter to false or the "Use FIPS compliant algorithms for encryption, hashing and signing" group policy to Not configured. After disabling FIPS mode, you can continue working at your current domain functional level in non-FIPS mode by restarting the agent:

/usr/share/centrifydc/centrifydc restart

If you want to enable FIPS mode, leave the current domain, update your domain functional level, then join a Windows Server 2008, or later, domain.

Importing sudoers configuration files

If you are currently managing privileges on Linux and UNIX computers using multiple sudoers configuration files, you can import that information and convert it into rights and role definitions that can then be assigned to Active Directory users and groups, local users and groups, or both.

This chapter describes how to migrate all of your privilege management information from sudoers configuration files to Active Directory through Access Manager.

Identify the sudoers file on each computer

Most organizations use sudoers configuration files and the sudo program to manage privileges on Linux and UNIX computers. To read the sudoers file on each Linux or UNIX computer, you must have root-level permission. You can define a command right to grant this level of access to other users.

The default location for the sudoers configuration file is /etc/sudoers, and in general, this is the file to import from each computer. However, there are some exceptions:

- If sudo was compiled with the --sysconfdir option to specify a different location for sudoers file, you need to find the actual location. Run sudo -V to see the sudo configuration options, including the path to the sudoers file.
- If your environment has an automatic mechanism for distributing a single sudoers file to the entire network, you can use that one file and don't need to import multiple files.

Get the sudoers file from each computer

You can manually copy the sudoers file to a location on the Windows computer that has Access Manager installed. You should specify a file name that identifies the computer where the file was used. For example, you might include the local host name or a functional description, such as oracle_server_sudoers.txt or qa1_server_sudoers.txt.

Import the sudoers file

After you have copied the sudoers file to the computer where Access Manager is installed, you can import the sudoers file into a selected Centrify zone.

To import the sudoers file

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you want to import the sudoers file.

   In most cases, if you have a sudoers file that covers multiple computers, you should import it into a parent zone so that it is available to multiple child zones. If the file is used on a single computer, you might select the specific child zone that contains that computer.
3. Right-click, then select **Import sudoers file.**
4. Click **Browse** and navigate to the location in which you copied the sudoers file, select the file, click **Open**, then click **Next**.
5. Review the contents of the file to verify you are ready to import, then click **Next**.

   If you have previously imported a sudoers file—for example, from a different computer—importing a new sudoers file overwrites the data from the previous import. If you have not yet converted the previous sudoers information to rights and rights in Access Manager, click **Cancel** to exit the wizard.

   For more information about convert the imported information to Centrify rights and roles, see Converting sudoers aliases and user specifications before importing an another sudoers file.
6. Review the parsing summary for errors or warnings to verify whether you are ready to import, then click **Next**.

   You can click **Details** to see the list of error and warnings, if applicable. From the list, you can select a specific error or warning, then click **Go To** to see the definition in the sudoers file. You can continue with the import if the list only displays warnings. If there are errors, you must fix them before continuing. Make note of any errors and warnings to fix, then click **Close** to close the Details list.

   If the file contains errors, or if you want to fix warnings before importing, click **Cancel** to exit the wizard. You can then open the sudoers file in a text editor to fix, delete, or comment out the lines in the file, then save it. After you have modified the file, you can rerun the Import Sudoers File wizard.
7. Click **Finish** to complete the import.

The import wizard creates a new node called Sudoers, which contains sub-nodes for the types of data contained in a sudoers file. For example, expand **Sudoers** to see the nodes for User Alias, Runas Alias, Host Alias, Command Alias, and User Specifications. If the Sudoers node is not visible, select Authorization, right-click, then click **Refresh**.

Some or all of the Sudoers sub-nodes might be empty depending on whether the sudoers file included definitions of that type. For example, if there are no user aliases defined in the sudoers file, the User Alias sub-node is displayed in Access Manager, but there are no entries under it.

You can now convert the sudoers data to rights, role definitions, and role assignments in the Centrify zone. If you intend to import more than one sudoers file into the same zone, you must convert the imported aliases and user specifications to rights, role definitions, and role assignments before importing another sudoers file.

Converting sudoers aliases and user specifications

Before you convert the sudoers file aliases and user specifications to rights, role definitions, and role assignments, be certain that you have imported all the users and groups specified in the sudoers file into Active Directory, and that you have added them to the zone in which you are imported the sudoers file. If there are users and groups without a profile in the zone when you attempt to convert the user specifications from the imported sudoers file into role assignments in Access Manager, the conversion will fail.

In addition, keep in mind that the role definitions and assignments you create from sudoers specifications do not contain any UNIX system rights or PAM access rights. You can assign those rights through other roles, such as the predefined UNIX Login role, or you can add system rights and PAM access rights to the role definitions after you create them from the sudoers specifications.

Within each item are objects for the sudoers definitions that were imported. For example, within User Alias are alias definitions, each one of which contains the user accounts defined for that alias.

Each type of information from the sudoers file converts to a different type of authorization information in the Centrify zone. You do not need to convert all of the imported aliases. You can simply ignore or delete aliases that are obsolete or no longer relevant.

Converting user aliases

On Linux and UNIX computers, a user alias in the sudoers file defines a set of users without creating a group. When you convert a user alias specification to be used in a zone, however, it becomes an Active Directory group. Assigning users to groups simplifies user management because if users change roles or leave the company, you can simply remove their group membership, without deleting their accounts, and effectively, they no longer have access to the roles assigned to members of the group.

You can create a new Active Directory group from the user alias you imported or map the imported alias to an existing Active Directory group.

To create a new Active Directory group from a user alias

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the sudoers file.
3. Expand Authorization and Sudoers, then select **User Alias**.
4. Select the alias name, right-click, then select **Create AD Group**.
5. Verify the container location, or click **Browse** to select a different container, then click **Next**.
6. Verify the group name, which defaults to the alias name, optionally, add a prefix or suffix, and select the scope for the group, then click **Next**.
7. Review the group and group membership information displayed, then click **Next**.

   If there are any warnings or errors displayed, you must fix the errors before continuing. If only warning are displayed, you can continue to create the group. For example, if the user alias has members that don't have a corresponding Active Directory account, you can continue creating the group.
8. Review information about the new Active Directory group, then click **Finish** to create the group.

To map a user alias to an existing group

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the sudoers file.
3. Expand Authorization and Sudoers, then select **User Alias**.
4. Select the alias name, right-click, and select **Map to AD Group**.
5. Select **Remove original AD group membership** or cancel the selection depending on whether you want to keep the current members of the group when adding the users from the alias definition.

   If you select this option, the wizard removes the existing members of the group when adding the new members. If you do not select this option, the wizard adds the new members to the existing members.
6. Click **Browse**, then enter search criteria to identify the group and click **Find Now**.

7. Select the name of the group and click **OK**.

   The wizard imports the users defined by the alias into the specified Active Directory group. It also issues a warning message that it can't import users who are defined by the alias but who are not defined in Active Directory.

Viewing run-as aliases

A run-as alias defines a group of one or more users who other users are able to run commands as. Select and double-click the alias name to expand it and see the users who are defined for it. You cannot directly import run-as aliases. However, if a user specification includes a run-as alias, you can view the run-as definition in the **Runas Alias** node, and import the commands defined in the specification. For more information about user specifications, see Converting user specifications.

Converting host aliases

Host alias definitions are popular in centralized sudoers files because they allow you to assign privileges to groups of computers rather than managing privileges on an individual computer and file basis. They convert naturally to computer roles, which also assign privileges to groups of computers.

When you convert a host alias to a computer role, the wizard creates a new computer role, creates an Active Directory group that contains the computers defined in the host alias, and adds these computers to the new computer role. Because the computer role group is an Active Directory group, the computers can span multiple zones and include computers that are joined to different zones. To complete the computer role definition, you must add the appropriate user role assignments, which specify what specific users and groups in different role definitions are allowed to do on the computers included in the computer role group.

To create a computer role from a host alias

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the sudoers file.
3. Expand Authorization and Sudoers, then select **Host Alias**.
4. Select the alias name, right-click, then select **Create Computer Role**.
5. Click **Next** to accept the location for the group of computers, or change the location, then click **Next**.
6. Verify or change the group name, optionally, add a prefix or suffix, and select the scope for the group, then click **Next**.
7. Review the group and group membership information displayed, then click **Next**.
8. Review information about the new Active Directory group for computers, then click **Finish** to create the group and the new computer role.

   If the computer accounts exist in Active Directory, the computers defined in the host alias are automatically added to the new Active Directory computer group and to the "Members" node of the new computer role.
9. Expand Authorization, Computer Roles, and the computer role name.
10. Select Role Assignments, right-click, and click **Assign Role**.
11. Select the role and click **OK**.
12. Click **Add AD Account**.
13. Select User or Group, enter search criteria, then click **Find Now** to search for and elect the user or group, then click **OK**.
14. Select the appropriate user or group from the result, then click **OK** to complete the user role assignment.

Viewing command aliases

You can select the Command Alias sub-node to view the command aliases that were imported from the sudoers file. You can't edit or delete the command aliases. The information is displayed for your reference. You can assign the command aliases listed role definitions, role groups, and computer roles when you convert the user specifications imported from the sudoers file.

Converting user specifications

In the sudoers file, user specifications make use of the alias definitions to assign commands and privileges to users. After you import the sudoers file, you can convert the user specifications into role assignments.

To convert user specifications to role definitions and role assignments

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the sudoers file.
3. Expand Authorization and Sudoers, then select **User Specifications**.
4. Select the name of a user specification, right-click, then select **Import**.
5. Review the list of commands to be created, then click **Next**.
6. Verify the name of the role definition name to be created, then click **Next**.

   By default, the role definition is named Role_n. You can change it after it is created.
7. If the user or group defined in the imported user specification is not found in the zone, the role assignment to be created is displayed and you can click **Next**, then click **Finish**.

   If the user or group defined in the imported user specification is not found in the zone, the role assignment will fail and the role displays an error (Error). Click **Cancel** to exit the wizard and add the user or group to Active Directory and the zone.

   Importing a user specification will fail if the user or group defined in the user specification is not found in the zone or if no computers are defined for the host alias in the user specification are found in the zone.
8. Rename the role definition by expanding Authorization and Role Definitions.
   - Select the new role definition, for example, Role_2.
   - Right-click, then select **Rename**
   - Type a new name for the role definition.

The role definitions you create from a sudoers specification do not contain the UNIX system rights or PAM access rights. You can assign these rights through a separate role assignment or by add the appropriate UNIX system rights and PAM access rights to the new role definitions.

Removing imported sudoers information

Once you have validated the conversion of imported sudoers file information, you can purge the sudoers information from Access Manager.

To purge sudoers information

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the sudoers file.
3. Expand Authorization, then select **Sudoers**.
4. Right-click, then select **Purge**.

   The Sudoers node and sub nodes are removed from Access Manager.

Mapping sudo to dzdo

To execute privileged commands users must type dzdo and the command name. If you want, you can map sudo to dzdo, which allows your users, who are accustomed to using sudo to execute their privileged commands, to continue to type sudo commandName. If the user has a role assignment that allow him to execute the command in an unrestricted shell, the command is executed using dzdo commandName. To map sudo to dzdo on computers in your organization, you can enable the "**Replace sudo by dzdo"** group policy for a site, domain, or organizational unit.

For more information about working with group policies, see the *Group Policy Guide*.

Using Centrify OpenLDAP proxy service

This section describes the Centrify OpenLDAP proxy service (centrifydc-ldapproxy) that you can use to map Active Directory users to UNIX identities to enable access to the files stored on legacy network appliance servers and storage devices. Centrify OpenLDAP also enables Linux and UNIX computers to search Active Directory domain controllers and global catalog servers for any information stored in Active Directory. If you have the appropriate permissions, you can also use the Centrify OpenLDAP proxy service to add, modify, or delete information stored in Active Directory.

What the OpenLDAP proxy provides

Many applications support the Lightweight Directory Access Protocol (LDAP) and require data stored in this format, but do not support Kerberos. In addition, many applications that support LDAP cannot search Active Directory directly because of the complexities of the Active Directory environment itself, such as the global catalog, multiple domains, multiple forests, and trust relationships.

The Centrify OpenLDAP proxy is an OpenLDAP server process that enables LDAP clients that are not Kerberos-enabled to search Active Directory efficiently and securely. By using the Centrify OpenLDAP proxy, applications that support LDAP can search complex Active Directory environments and authenticate users with Active Directory. Through the Centrify Agent, the Centrify OpenLDAP proxy enables you to resolve UID, GID, and group membership efficiently and collapse the entire Centrify hierarchical zone structure, including parent and child zone, and individual computer overrides into a single namespace for LDAP applications.

In addition, connecting to Active Directory typically requires an authenticated bind with a valid user name and password. Because the Centrify OpenLDAP proxy uses the Centrify Agent to connect to Active Directory and retrieve information, you can issue OpenLDAP commands without an authenticated bind.

The following diagram provides a simplified overview of the components.

Components

The key advantages to deploying the Centrify OpenLDAP proxy when you have LDAP clients where the Centrify Agent cannot be installed are as follows:

- You can use the Centrify OpenLDAP proxy server to run commands that retrieve or update information stored in Active Directory.
- The Centrify OpenLDAP proxy service uses the Centrify Agent to securely connect to Active Directory and retrieve user, group, and other information from the Active Directory domain controller.
- You can leverage the offline authentication and caching capabilities of the Centrify Agent for applications that support LDAP, but not Kerberos.
- Regardless of the complexity in Active Directory, including multiple domains and forests and parent and child zones, the Centrify OpenLDAP proxy treats the information stored in Active Directory as a single RFC2307-compatible namespace.

Enabling simple authentication

Users can be authenticated through simple authentication to the Centrify OpenLDAP proxy with their username and password. This is then converted to a secure Kerberos authentication by adclient.

By default, to authenticate users, adclient checks its credential cache data first, then, if not in cache, it refers to Active Directory. Allowing Centrify OpenLDAP proxy to use the adclient credential cache, enables authentication if adclient is in disconnected mode.

If you want to always authenticate through Active Directory:

To the slapd.conf file:

/etc/centrifydc/openldap/slapd.conf

Add:

cdc-auth-prefer-cache false

Enabling simple proxy mode

If either objectClass or objectCategory is not specified in the search filter, the search is in simple proxy mode. With simple proxy mode, all search filters are sent without translation through adclient to Active Directory. All results are returned as provided by Active Directory without translation or interpretation of results.

Accessing network appliance or storage servers

One of the most common uses for the Centrify OpenLDAP proxy service is to provide access to the files stored on legacy network appliance file servers and storage devices. Many organizations use network appliance file servers and storage area network devices to provide highly available and scalable data storage services that support multiple client access protocols—including NFS, CIFS and iSCSI—and multiple operating systems.

Supporting multiple protocols and operating systems, however, presents a challenge when users want to access files from computers with different operating systems. To ensure users are granted proper access to files stored on a network appliance or storage server, their identity attributes must be consistently defined for both UNIX and Windows operating systems.

For example, the identity attributes that allow access to the files on a network appliance or storage server might be UNIX profile attributes from a common NIS or LDAP repository. The UID and GID values establish file ownership and file access permissions. For Windows users to access the files stored on the network appliance or storage server, their Windows account must be mapped to the UNIX profile that grants them the appropriate file permissions.

Delinea

Mapping Active Directory users to UNIX profiles

Centrify enables you to map Active Directory users to one or more UNIX profiles. The UNIX profile contains each user's identity attributes. You can use this mapping of Active Directory account information to UNIX identity attributes to provide consistent file and directory ownership and access rights to files that are stored on a network appliance or storage server.

By mapping an Active Directory account to a UNIX profile, you can ensure that a user's identity is consistently maintained and that access to UNIX-hosted resources is properly protected regardless of the computer from which the user accesses the resource.

For network appliance or storage servers that are hosted on UNIX computers and require UNIX identity attributes to grant access, you can use the Centrify OpenLDAP proxy service to make the Active Directory-hosted user mapping information available through the LDAP or LDAPS protocol.

Configuring servers to use the proxy service

Before you can use the Centrify OpenLDAP proxy service to look up information stored in Active Directory, the network appliance, storage device, or file server you want to use must be configured to use LDAP to look up user and group information. In most cases, this is an option you configure when setting up a server or device.

If your vendor supports connecting to LDAP servers for authentication and authorization services, configuring the server or device to use the Centrify OpenLDAP proxy requires the following high-level steps:

1. Install Access Manager, create at least one zone, and add users to the zone.
2. Install the Centrify Agent on a Linux or UNIX computer and join the computer to an Active Directory domain.
3. Install the centrifydc-ldapproxy package on the Linux or UNIX computer.
4. Start the centrify-ldapproxy service and verify proper operation.
5. Set up the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service to look up user and group information.
6. Test the solution for proper end-to-end operation.

Installing the Centrify OpenLDAP proxy service

On most platforms, the centrifydc-ldapproxy package is available with the Centrify agent software package but is not installed by default. You can select the package in the installation script or install it using a native package installer.

To run the Centrify OpenLDAP proxy service, the computer must:

- Be joined to an Active Directory domain.
- Have the Centrify Agent installed and the adclient running.

In the following example, the agent is installed on a Linux computer and the computer is joined to the acme.org Active Directory domain.

To install the Centrify OpenLDAP proxy service on a Linux computer

1. Log on or switch to the root user, then navigate to the directory where you extracted Centrify files.

   For example, if you ran the gunzip and tar commands in the /tmp directory, change to the /tmp directory.

2. Run install.sh or a native package manager to install the files.

   For example, run the following command:

   ./install.sh

   You can type K to keep any existing packages you have installed. When you see the Install the CentrifyDC-ldapproxy package prompt, type Y. Follow the remaining prompts displayed to complete the installation.

   Alternatively, you can use a native package manager. For example on most Linux distributions, you can run a command similar to this:

   rpm -Uvh centrifydc-ldapproxy-release-arch.rpm

   If you are installing on Solaris, unzip and extract the contents of the package, then run a command like this:

   pkgadd –d CentrifyDC-ldapproxy -a admin

   If you are using an installation program, such as SMIT or YAST, see the documentation for that program.

3. If you want to start the ldapproxy service with parameters, configure the STARTUP-OPTS option.

   Run the appropriate command for your platform.

   - For CentOS, SLES

echo "STARTUP_OPTS=\"-h ldaps:///\"" >> /etc/sysconfig/centrify-ldapproxy

- For Debian

  echo "STARTUP_OPTS=\"-h ldaps:///\"" >> /etc/default/centrifyldapproxy

- For HPUX

  echo "STARTUP_OPTS=\"-h ldaps:///\"" >> /etc/rc.config.d/centrify-ldapproxy

- For AIX

  chssys -a "-d 0 -h ldaps:///" -s centrify-ldapproxy

- For Solaris without Service Management Facility (SMF)

  echo "STARTUP_OPTS=\"-h ldaps:///\"" >> /etc/centrifydc/openldap/centrify-ldapproxy.conf

- For Solaris with Service Management Facility (SMF)

  svccfg -s centrify-ldapproxy setprop 'slapd/STARTUP_OPTS=("-h""ldaps:///")'

4. Start the centrify-ldapproxy service.

   For example, on Linux computers:

   /usr/share/centrifydc/bin/centrify-ldapproxy start

5. Test the service by searching for an object in the Active Directory domain.

   For example, to search for groups in the domain, you might type commands like this:

   ```
   cd /usr/share/centrifydc/bin
   ldapsearch -h localhost -p 389 -x -b “dc=acme,dc=org”
   s sub "objectClass=group" -D
   "cn=amy.adams,cn=users,dc=acme,dc=org" -w 1234abcepassword
   ```

   The -h and -p options are required to connect to Active Directory using the proxy service and the Centrify Agent. If the LDAP proxy service is not on the local computer, use the -h option to specify the name of the computer where you have installed it.

   You can also connect to Active Directory directly using a valid user name and password. For example:

   ```
   ldapsearch -D "cn=amy.adams,cn=users,dc=acme,dc=org" -W
   -h dc2012.acme.org -p 389 -x -b "dc=acme,dc=org"
   -s sub "objectClass=group"
   ```

6. (Optional) Review and modify, if necessary, the default centrifyldapproxy service start-up script in the /etc/init.d/ directory.

   You can use the /usr/share/centrifydc/bin/centrifyldapproxy script to start, stop, restart or check the status of the Centrify OpenLDAP proxy service.

   Note: By default, the service starts automatically when the computer restarts.

Specifying the LDAP server

After you have installed and tested the Centrify OpenLDAP proxy service, the next step is to configure the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service to look up user and group information. In most cases, this involves setting configuration options to specify the computer where the Centrify OpenLDAP proxy service is running as the LDAP server you want to use in a local or system-wide ldap.conf file. You should consult the documentation provided by the vendor you are integrating with for details about how to set up LDAP integration.

Testing the solution

After you have configured the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service on a Centrifymanaged computer, you should verify that files created by a Windows user have the correct UID and GID to access those files from both a UNIX computer and a Windows computer.

Delinea

Manually starting the OpenLDAP service

Typically, the Centrify OpenLDAP proxy service automatically starts when the computer restarts. You have the option to manually start Centrify OpenLDAP proxy service.

The Centrify OpenLDAP proxy service is a modified version of the standard slapd LDAP server process. The Centrify version of the slapd process also uses a customized version of the standard LDAP server configuration file in the following location:

/etc/centrifydc/openldap/slapd.conf

This customized version of the slapd.conf configuration file is created automatically when you join a domain and is configured by default with Access Manager and domain-specific information. You can start or stop the slapd process at any time by using the centrify-ldapproxy script or directly from the command line.

To start the Centrify OpenLDAP proxy service directly from the command line using the default configuration file, you can run the following command:

/usr/share/centrifydc/libexec/slapd

or

/usr/share/centrifydc/bin/centrify-ldapproxy start

If you start the slapd process directly from the command line, you can also specify additional command line options just as you would for the standard slapd LDAP server process. For example, you can use the -f command line option to specify a different configuration file to use:

slapd -f /etc/centrifydc/openldap/slapd.conf

or

/usr/share/centrifydc/bin/centrify-ldapproxy start -f /
etc/centrifydc/openldap/slapd.conf

For more information about the command line options when starting the LDAP server directly from the command line, see the man page for the slapd process.

Note: On the computer that is using systemctl, if the slapd process crashes, the systemd process will restart the slapd process.

Sample deployment scenario

The following diagram illustrates a basic deployment scenario for a distributed site with minimal OpenLDAP proxy overhead. As depicted in this illustration, the legacy LDAP servers and devices support a redundant LDAP server for fault tolerance and failover.

Using OpenLDAP commands

The Centrify OpenLDAP proxy service includes a set of OpenLDAP commands that have been modified to support looking up information in Active Directory domain controllers and the global catalog. The Centrify distribution of OpenLDAP supports most of the standard options and syntax for performing LDAP operations, but the ldap commands in the Centrify distribution of OpenLDAP also support the following options that are not supported in a standard OpenLDAP distribution:

-m	Use the local machine credentials from the /etc/krb5.keytab file. This option requires root user access.
-r	Disable line wrapping when printing out LDIF entries.

The Centrify distribution of OpenLDAP also provides extended URL support for Active Directory. With Centrify LDAP commands, you can use the following URLs to connect to Active Directory computers:

ldap://domain_name	Connect to the appropriate domain controller for the specified domain within the Active Directory site.
ldap://	Connect to the joined domain.
gc://[domain_name]	Connect to the global catalog domain controller for the joined domain. You can use the optional domain_name parameter to specify a domain in a different forest.

The Centrify distribution of OpenLDAP includes the following commands:

- ldapsearch
- ldapadd
- ldapmodify
- ldapmodrdn
- ldapcompare
- ldapdelete

Note: The ldappasswd and ldapwhoami commands do not work with Active Directory. For more information about using the OpenLDAP commands or the standard options available, see the man page for each command.

Centrify OpenLDAP proxy commands attributes

The Centrify OpenLDAP proxy commands accept the following attributes.

- dn - Specifying the dn attribute returns only the distinguished name
- 1.1 - Specifying the 1.1 attribute returns only the distinguished name
- * - Specifying the asterisk (*) attribute return is situational:
  - If only * is specified, Centrify OpenLDAP proxy returns all our supported attributes.
  - If the * is specified with additional attributes, Centrify OpenLDAP proxy returns the given additional attributes.

Searching for users and groups

If you want to use ldapsearch to find a user, do not use objectclass=user or objectcategory=person to specify the filter. Instead, you should use objectclass=posixaccount. For example, to find the user with the UNIX name jtr enter a command similar to the following:

/usr/share/centrifydc/bin/ldapsearch -x -h localhost -D
“CN=Administrator,CN=Users,DC=pistolas,DC=org” -W -b
“dc=pistolas,dc=org” "(&(objectclass=posixaccount)(uid=jtr))"“

Optionally, use the UID number instead of the UNIX name:

"(&(objectclass=posixaccount)(uidNumber=1234567))"

Similarly, use objectclass=posixgroup to retrieve information on a group. This filter supports the following options:

- cn: Find a group with a given UNIX name
- gidNumber: Find a group with a given GID
- memberUID: Search for secondary group membership of given UNIX user.

Searching the global catalogs

In most cases, you use the Centrify OpenLDAP proxy service to search for information through the domain controller. However, you can also use the Centrify OpenLDAP proxy service to perform searches in the global catalog, if needed. The global catalog search is especially useful if you have a large, multiple-domain forest.

To specify that you want the Centrify OpenLDAP proxy service to search the global catalog, add “CN=$” to the front of the search base.

To search Active Directory for a specific account, use the syntax:

"(&(objectCategory=Person)(Name=amy.adams*))"

For example, in the global catalog, you might type a command similar

to the following:

/usr/share/centrifydc/bin/ldapsearch -h localhost -D
"cn=amy.adams,cn=NewUsers,dc=ajax,dc=org" -w password -x -b "cn=$"

By default the Centrify OpenLDAP proxy service is configured to disable anonymous binds. To allow anonymous binds:

1. Edit the /etc/centrifydc/openldap/slapd.conf file.
2. Remove or comment following line.

   require authc

If anonymous binds are disabled, you no longer need to specify the -D and -w parameters to invoke an ldapsearch. For example:

ldapsearch -h localhost -x -b "dc=wonder,dc=land"
"(&(objectClass=User)(displayName=Mister*))" displayName

Minimizing search traffic to adclient

To minimize the traffic to adclient and subsequently to Active Directory, during an ldapsearch, the Centrify OpenLDAP proxy implements memory cache. The Centrify OpenLDAP proxy memory cache is disabled by default.

To enable the Centrify OpenLDAP proxy memory cache, change slapd.conf to:

ldapproxy.cache.enabled true

Enabling encrypted communication

By default, communication between LDAP clients and the Centrify OpenLDAP proxy service is not encrypted. To secure communications between LDAP clients and the Centrify OpenLDAP proxy service using Transport Layer Security (TLS), you must create or obtain the required certificates and configure both the LDAP client and the LDAP server to use the certificates. In addition, you must configure the LDAP server with the certification authority (CA) certificate, its own server certificate, and a private key.

The current versions of the ldapsearch client and ldapproxy server support Transport Layer Security (TLS) v1.2.

Depending on your network topology, you might also need to modify client-side or server-side configuration settings to successfully return search results.

Preparing for auto-enrollment

You can configure the Centrify OpenLDAP proxy service to automatically get the certificate, private key, and CA chain for secure LDAP (ldaps) connections. To configure automatic enrollment for certificates, however, you must have an Active Directory domain controller that you can use as a certification authority for issuing certificates.

The following steps summarize how to prepare the domain controller:

1. Use Server Manager to add the Active Directory Certificate Services role to a domain controller.
2. In the Add Roles wizard, select the Certification Authority role service and follow the prompts displayed to configure the server role.
3. Open the Certificates MMC snap-in, select the domain controller certificate, right-click, then click Open.
4. Select the Details tab, click Copy to file, then follow the prompts displayed to export the certificate to a file.
5. From Administrative Tools, select Group Policy Management, then select an appropriate Group Policy Object for the forest and domain you want to edit.
6. Right-click the Group Policy Object, then click **Edit**.
7. Under Computer Configuration, expand Policies > Windows Settings > Security Settings, then select Public Key Policies.
8. Select Trusted Root Certificate Authorities, right-click to select **Import**, then follow the prompts displayed to import the certificate.
9. Select Certificate Services Client - Auto-Enrollment, then select **Enabled**.
10. From Administrative Tools, select Certification Authority, expand the name of the domain controller you are using as the certification authority, then select **Certificate Templates**.
11. Right-click to select Manage, select an appropriate template to use, such as the Computer template, right-click, then click **Duplicate Template** to open the properties page for the new template.
12. Type an appropriate name for the new template, such as Centrify OpenLDAP Proxy.
13. Click the Security tab, select the Domain Computers group, select Allow for the Autoenroll permission, then click **Apply**.

    You can set other properties on the remaining tabs, as needed. For example, you might want to click the Subject Name tab to change the subject name format to Fully distinguished name. When you are finished setting properties for the template, click **OK**.
14. In the Certification Authority console, select Certificate Templates, right-click to select New, then click **Certificate Template to Issue**.
15. Select the template you created, for example, select the Centrify OpenLDAP Proxy template, then click **OK**.

Updating the Centrify OpenLDAP proxy computer

After you have prepared the domain controller with the policy for certificate autoenrollment, you can use the following steps to provide the required certificate, private key, and certification authority.

1. Verify the computer where you are running the Centrify LDAP proxy service is joined to an Active Directory domain.
2. Change to the directory where certificates for auto-enrollment are located.

   cd /var/centrify/net/certs/

   You should see files similar to the following listed in the directory:

   auto_LDAPProxy.cert
   auto_LDAPProxy.chain
   auto_LDAPProxy.key
   trust_41DFF689876FCE52E02EE73FC7E3782964DC54BB.crl
   trust_F7842B2A65489F15A1722518E41F5E6B0F4FBC5E.cert

3. Run an openssl command similar to the following to create the certificate:

   openssl pkcs7 -in auto_LDAPProxy.chain -text -out auto_LDAPProxy_CA.pem print_certs

4. Add the following lines to /etc/centrifydc/openldap/slapd.conf configuration file. Comment out the old TLSCipherSuite line, as shown here.

   TLSCertificateFile /var/centrify/net/certs/auto_LDAPProxy_CA.pem
   TLSCertificateFile /var/centrify/net/certs/auto_LDAPProxy.cert
   TLSCertificateKeyFile /var/centrify/net/certs/auto_LDAPProxy.key
   TLSCipherSuite TLSv1.2
   # TLSCipherSuite SSLv3

   You should also review and modify other server configuration settings, if needed. For example, you might use settings similar to the following:

   # Require START TLS on port 389
   security tls=1
   # Require TLS v1.0 or better
   TLSProtocolMin 3.1
   TLSVerifyClient try

5. Add the following line to /etc/centrifydc/openldap/ldap.conf configuration file:

   TLS_CACERT /var/centrify/net/certs/auto_LDAPProxy_CA.pem

   You should also review and modify other configuration settings, if needed. For example, you might need to change the TIMEOUT value to allow clients to wait an appropriate number of seconds for a response:

   TIMEOUT 15

6. Restart the Centrify OpenLDAP proxy service.

   sudo /usr/share/centrifydc/bin/centrify-ldapproxy start -h ldaps:///

7. Test operation by running an OpenLDAP command, such as ldapsearch.

   /usr/share/centrifydc/bin/ldapsearch -x -H ldaps://localhost:636 -b 'cn=users,dc=win2012,dc=test' -D administrator@win2012.test -W "(cn=test_user)"

8. To confirm that TLSv1.2 is being used, use openssl s_client to connect to the slapd. For example, enter:

   $ openssl s_client -connect localhost:636 -showcerts -state -CAfile /etc/centrifydc/openldap/cacert.pem

9. Review the output from the previous command and confirm that the protocol is TLSv1.2, as shown here:

   ...
   SSL Session:
   Protocol : TLSv1.2

10. (Optional) Alternatively, to confirm that TLSv1.2 is used, run a software tool like Wireshark to capture and inspect the ldapsearch traffic.

Securing communication without auto-enrollment

If you are not using an Active Directory domain controller and autoenrollment for certificate distribution, you can manually configure the Centrify OpenLDAP proxy service to use the server certificate and private key you create.

The following steps summarize how you can manually configure the Centrify OpenLDAP proxy service to use certificates.

1. Use CA.sh to create the certificates:

   /usr/share/centrifydc/ssl/misc/CA.pl -newca

   /usr/share/centrifydc/bin/openssl req -new -nodes -keyout newreq.pem -out newreq.pem

   /usr/share/centrifydc/ssl/misc/CA.pl -sign

2. Install the certificates in the /etc/centrifydc/openldap directory.

   cp demoCA/cacert.pem /etc/centrifydc/openldap/cacert.pem

   mv newcert.pem /etc/centrifydc/openldap/servercrt.pem

   mv newreq.pem /etc/centrifydc/openldap/serverkey.pem

3. Add the following lines to /etc/centrifydc/openldap/slapd.conf configuration file:

   TLSCACertificateFile /etc/centrifydc/openldap/cacert.pem

   TLSCertificateFile /etc/centrifydc/openldap/servercrt.pem

   TLSCertificateKeyFile /etc/centrifydc/openldap/serverkey.pem

4. Add the following line to /etc/centrifydc/openldap/ldap.conf configuration file:

   TLS_CACERT /etc/centrifydc/openldap/cacert.pem

5. Start the slapd deamon using the following:

   /usr/share/centrifydc/libexec/slapd -h "ldaps:///"

   or

   sudo /usr/share/centrifydc/bin/centrify-ldapproxy start -h ldaps:///

Searching for automount maps and entries

You can use the Centrify ldapproxy service and ldapsearch command to find automount maps and automount map entries that you have stored in Active Directory. The following examples illustrate how to write filters to retrieve automount information. These examples assume you have added the following automount maps and map entires to Active Directory using Access Manager:

- The auto.home map has the map entries test1 and test2.
- The autotest map has the map entries test10 and test11.

To retrieve both maps, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test" "(objectClass=automountMap)"
-D "cn=amy.adams,cn=users,dc=acme,dc=org" -w 1234abcepassword
```

The command returns attribute information for each map similar to this:

```
dn: cn=auto.home,cn=NisMaps,cn=global,cn=Zones,dc=acme,dc=test
automountMapName: auto.home
ou: auto.home
cn: auto.home
displayName: $CimsAutomountMapVersion1
objectClass: top
objectClass: automountMap
uSNChanged: 20046
```

To retrieve information for a specific map, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test"
"(&(objectClass=automountMap)(automountMapName=auto.home))"
-D "cn=amy.adams,cn=users,dc=acme,dc=org" -w 1234abcepassword
```

To retrieve all map entries from both maps, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test" "(objectClass=automount)"
-D "cn=amy.adams,cn=users,dc=acme,dc=org" -w 1234abcepassword
```

To retrieve information for a specific map entry, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "cn=auto.home,DC=acme,DC=test"
"(&(objectClass=automount)(automountKey=test1))"
-D "cn=amy.adams,cn=users,dc=acme,dc=org" -w 1234abcepassword
```

For information about adding and managing NIS maps to Active Directory, see the *Network Information Service Administrator's Guide*.

Automatic translation to search for zone users

If you integrate the Centrify Agent with a software environment that has limited configuration options, a standard ldapsearch query might fail to return zone users and groups. If you encounter this issue, you can use a configuration parameter to automatically translate a standard search for Active Directory users and groups into a search query for zone users and groups.

You can set the ldapproxy.cdctranslate.fetchbydnuid parameter in the slapd.conf configuration file to true if you want a search for Active Directory users and groups to be automatically translated into a search for zone users and groups. The default is false. After changing the parameter setting, you should restart the centrify-ldapproxy service.

Note that the translation only applies if the ldapproxy.cdctranslate.fetchbydnuid parameter is set to true, and the following additional conditions are in the search request:

- For the search base, the first part of the DN must be "uid=unixname"
- The search scope base must be (0)
- The search filter must be (objectClass=*)

For example, automatic translation is performed if you run a command similar to the following after changing the ldapproxy.cdctranslate.fetchbydnuid parameter to true and restarting the centrify-ldapproxy service:

```
ldapsearch -x -D "cn=zoe,OU=ajax,dc=acme,dc=org" -w 1234abcepassword
-h localhost "(objectClass=*)" -b "uid=zoe,OU=ajax,dc=acme,dc=org"
-s base
```

Using workstation mode and Auto Zone

For most organizations, adding Linux or UNIX computers to an Active Directory domain involves creating one or more parent zones, adding Active Directory users and groups to the zone, and assigning one or more roles to the zone users. As an alternative to this process, you can create a single Auto Zone for all Active Directory users or a specific subset of Active Directory users.

Profiles are generated for all users in the forest

If you use Auto Zone, all of the profile attributes that are normally defined in the zone to which a computer is joined are generated automatically based on user attributes in Active Directory or based on a set of agent configuration parameters. By default, all Active Directory users and groups in for the forest automatically become valid users and groups on the computers joined to Auto Zone. The generated profiles have a unique UID and GID for each Active Directory user in the forest. The generated profile is what enables access to the computers joined to Auto Zone.

In addition, if you have Active Directory users in another forest that has a two-way trust relationship with the forest of the joined domain, all of those users are also valid users for the joined computer.

Note: Auto Zone does not support one-way trusts. If a computer is joined to a domain that has a one-way trust relationship with another domain, the users and groups in the trusted domain do not become valid users and groups on the computer.

Limiting users and groups in Auto Zone

If you want to use Auto Zone, but do not want to give all Active Directory users and groups a valid profile, you can use group policies or configuration settings to limit the generation of profiles and computer access to specific users and groups. For example, if you have a two Active Directory groups with users who need access to Linux, UNIX, or Mac OS X computers, you can use either group policies or configuration settings to specify that only the two groups who require profiles are valid Auto Zone users and all other Active Directory users should be ignored.

Auto Zone does not provide zone-specific features

If you decide to use Auto Zone, you should keep in mind that Auto Zone does not support any zone-specific features, such as the ability to define rights and roles, assign roles to users and groups, configure auditing, or keep legacy identity attributes on different computers.

If you want to configure role-based access rights, delegate administrative activity, or migrate existing users and groups, you should not use Auto Zone. If you have a large Active Directory forest, but only require automatic profile generation for a subset of users and groups, you might want to use a combination of hierarchical zone and Auto Zone.

Joining a domain as a workstation

Auto Zone is created automatically in Active Directory if you join a domain by running the adjoin command with the --workstation option.

What to do before joining Auto Zone

Before joining a computer to Auto Zone, be certain that the following are true:

- Active Directory identities are unique for the forest and any two-way trusted forest.
- The Active Directory users and groups require a single set of properties for all computers that join the domain through Auto Zone and do not need to be segregated into zones for any reason.
- All domains in the forest and any trusted external forest must be unique or the join will fail. In this case, you must manually configure a unique prefix for each trusted domain using configuration parameters.

Who should perform this task

A Linux or UNIX administrator with root permission on the computers you want to join to an Active Directory domain. The administrator must also know the password for an Active Directory domain administrator account.

How often you should perform this task

In most cases, you only do this once for each Linux or UNIX computer that needs to join an Active Directory domain as a workstation.

Rights required for this task

You must have an account with root permission to modify agent configuration files on managed computers or an administrative account with write permission to enable group policies on a Group Policy Object linked to a domain or organizational unit.

Steps for completing this task

The following instructions illustrate how to join Auto Zone using the adjoin command.

To join a computer to a domain as a workstation

1. Log on the computer with the Centrify Agent using an account with root privilege.
2. Open a terminal and execute the following command:

   adjoin domainName --workstation

   For example:

   [root@rhe5]#adjoin acme.com --workstation
3. Type the Active Directory administrator's password.

   Administrator@ACME.COM's password:

   Using domain controller: win-f7d27u7kl6m.acme.com writeable=true
   Join to domain:acme.com, zone: Auto Zone succesful
4. Run the adinfo command to verify the connection to Auto Zone:

   [root@rhe5]# adinfo
   Local host name: rhe5
   Joined to domain: acme.com
   Joined as: rhe5.acme.com
   Pre-win2K name: rhe5
   Current DC: win-f72d7u7kl6m.acme.com
   Preferred site: Default-First-Site
   Zone: Auto Zone

Last password set: 2012-09-30 18:08:34 PDT
CentrifyDC mode: connected
Licensed Features: Enabled

Generating profiles for specific users and groups

You can automatically generate profiles for specific users and groups by enabling group policies in a Group Policy Object for a domain, site, or organizational unit in an Active Directory forest or by specifying configuration settings on individual computers.

Rights required for this task

You must have an account with root permission to modify agent configuration files on managed computers or an administrative account with write permission to enable group policies on a Group Policy Object linked to a domain or organizational unit.

Who should perform this task

A Windows or UNIX administrator performs this task, depending on your organization's policies. In most cases, a Windows administrator is responsible for configuring group policies and modifying Group Policy Objects. If your organization uses local configuration settings, the UNIX administrator is usually responsible for this task.

Steps for completing this task using group policies

In most cases, you should use group policies in a Group Policy Object to identify the Active Directory users and groups for which you want to automatically generate profiles. The Group Policy Object enables you to centrally manage access to computers in the Auto Zone. You can enable and configure the following group policies to specify a subset of Active Directory users and groups that should have access to computers in Auto Zone:

- Specify AD users allowed in Auto Zone
- Specify groups of AD users allowed in Auto Zone
- Specify AD groups allowed in Auto Zone

The following instructions illustrate how to limit the valid users and groups in the Auto Zone using these group policy settings.

To specify users and groups to include in Auto Zone by using group policy settings

1. Identify or create an Active Directory group that includes all of the users that you want to give access to Centrify-managed computers.

   The group can be a domain local, global, or universal group. The group can include sub groups — members of these sub groups will also be included in Auto Zone.

2. Open Group Policy Management to create or select a Group Policy Object that is linked to a site, domain, or organizational unit.
3. Right-click the Group Policy Object, then select **Edit** to open Group Policy Management Editor.
4. Expand Computer Configuration > Policies > Centrify Settings > DirectControl Settings, click Adclient Settings.
   - Double-click "Specify groups of AD users allowed in Auto Zone" to specify users by Active Directory group without automatically generating profiles for the groups themselves.
   - Double-click Specify AD users allowed in Auto Zone to specify individual Active Directory users for which to automatically generate profile.
   - Double-click Specify AD groups allowed in Auto Zone to specify individual Active Directory groups for which to automatically generate profile.
5. Select Enabled, then click List to browse for the groups or users to include.
6. Click **Add**, enter search criteria, then click **Find Now**.
7. Select one or more groups or users from the list, then click **OK**.

Steps for completing this task using configuration parameters

In some cases, you might want to limit the Active Directory users and groups who have a profile generated by configuring parameters in the centrifydc.conf file on individual computers. For example, you might want to use configuration parameter settings if you don't want to implement or apply group policies on certain computers.

You can configure the following configuration parameters to specify a subset of Active Directory users and groups that should have access to computers in

Auto Zone:

- auto.schema.allow.users
- auto.schema.allow.groups
- auto.schema.groups

The following instructions illustrate how to limit the valid users and groups in the Auto Zone using these configuration parameters settings.

To specify users and groups to add to Auto Zone by using configuration parameters

1. On a Windows computer, in Active Directory Users and Computers, identify or create a group or group that includes all the users who you want to have access to your Centrifymanaged computers.
2. On each computer to add to Auto Zone, open the /etc/centrifydc/centrifydc.conf configuration file.
    - Find the auto.schema.allow.groups parameter and remove the comment (#) to add the names of groups separated by commas.
    - Find auto.schema.allow.users and remove the comment (#) to add the names of users separated by commas.
    - Find auto.schema.groups and remove the comment (#) to add the names of groups separated by commas.

      The configuration file contains comments that list the valid formats for user and group names. For more information about setting these parameters or editing the configuration file, see the*Configuration and Tuning Reference Guide*.
3. Save and close the file.

Troubleshooting authentication and authorization

This chapter describes how to use diagnostic tools and log files to retrieve information about the operation of Server Suite software and how to identify and correct problems within your environment.

Diagnostic tools and log files

All Centrify services include diagnostic tools and logging mechanisms to help you trace the source of problems if they occur. These diagnostic tools and log files allow you to periodically check your environment and view information about Centrify operation, your Active Directory connections, and the configuration settings for individual UNIX and Linux computers.

Although logging is not enabled by default for performance reasons, log files provide a detailed record of Centrify Agent (adclient) activity. This information can be used to analyze the behavior of adclient and communication with Active Directory to locate points of failure. However, log files and other diagnostic tools provide an internal view of operation and are primarily intended for Centrify experts and technical staff.

In most cases, you should only enable logging when you need to troubleshoot unexpected behavior, authentication failure, or problems with connecting to Active Directory or when requested to do so by Centrify Support. Other troubleshooting tools, such as command line programs, can be used at any time to collect or display information about your environment.

Analyzing information in Active Directory

One important way you can troubleshoot your environment is by running the Analyze command. The Analyze command enables you to selectively check the integrity of the information stored in Active Directory. With the Analyze wizard, you can check for a variety of potential problems, such as duplicate user IDs, duplicate groups, empty zones, orphaned data objects, or computers that have joined more than one zone.

Note: When you run the Analyze command, only the zones that are open are checked.

To check for problems with information in the Active Directory forest:

1. Open Access Manager.

   If you are prompted to connect to a forest, specify the forest domain or domain controller to which you want to connect.

2. In the console tree, select the Access Manager root node, right-click, then click **Analyze**.

3. Select the types of checks you want to perform, then click **Next** to generate the report.

All	Perform all of the data integrity checks. **Note** If you do not register the administrative notification handler through the Setup Wizard or manually using ADSI, you should periodically run the Analyze command with **All** or **Orphan UNIX data objects** selected.
Computers joined to multiple zones	Check for computers that have joined the domain using more than one zone. Each UNIX computer should only reside in one zone, but if you run the join command more than once, it is possible to have the same computer in more than one zone. This option checks for this problem.
Cyclic zone hierarchy	Check for a circular zone hierarchy. The console prevents you from creating a circular zone hierarchy, but it is possible to do so inadvertently when using ADEdit.
Duplicate groups in zones	Check for duplicate UNIX group names or group identifiers (GIDs) in each open Centrify zone.
Duplicate role assignment containers in computer	Check for computers that have more than one location to store role assignment information.
Duplicate service principal names in forest	Check for duplicate service principal names across the entire forest. Service principal names are required to be unique within an Active Directory forest.
Duplicate SFU zones	Check for duplicate SFU zones that are set to manage the same NIS domain.
Duplicate users in zones	Check for duplicate UNIX user names or user identifiers (UIDs) in each open Centrify zone.
Duplicate zone default container	Check for duplicate Zones parent container objects in the Active Directory forest.
Empty	

computer roles	Check for computer roles that contain no computers or role assignments.
Empty profiles in hierarchical zones	Check for hierarchical zones that contain users or groups that have no profile data defined.
Empty zones	Check for zones that have no computers, users, or groups.
Foreign Security Principal Clean Up	Check for foreign security principal objects whose corresponding security principal has been removed.
Incomplete user UNIX data	Check for users with missing UNIX profile attributes or who are missing a primary profile. This analysis option checks the entire zone hierarchy for profiles with missing attributes and for users who have multiple profiles defined but don not have a primary profile. Users with an incomplete profile or a missing primary profile will not be able to log on even if they are assigned a role with login rights. Note that a profile can be incomplete at any level of the zone hierarchy as long as it is complete at the level where a computer is joined.
Inconsistency in granting NIS server permissions	Check that there is a zone_nis_servers group in each zone that supports agentless authentication and that the group contains all the NIS servers that have been defined for the zone. The zone_nis_servers group is required to assign permissions to managed computers that act as NIS servers, and should not be manually deleted or modified. This option checks that the group exists and includes all of the computers acting as NIS servers to ensure data integrity.
Inconsistent computer object names	Check for discrepancy between the DNS name for a computer in Active Directory and its Centrify computer profile name.
Insufficient permission for agent version update	Check whether the computer object in Active Directory has sufficient permission to update the version number property of the Centrify Agent for *NIX in the computer's serviceConnectionPoint object. If the computer object does not have permission to change this property, the version number cannot be displayed.
Insufficient permission for OS version update	Check whether the computer object in Active Directory has sufficient permission to update the version number property of the operating system in the computer's serviceConnectionPoint object. If the computer object does not have permission to change this property, the operating system version number cannot be displayed.
Invalid right assignments	Check whether an invalid right has been assigned to a role. This error occurs if a right has been added to a role and subsequently the right becomes invalid. Generally, a right becomes invalid if it is edited with a third-party tool, such as ADSI Edit, and an attribute is set to an invalid value. For example, Access Manager creates Active Directory objects of type msDSAZOperation for command- and PAM-application-rights, and assigns a HEX value to the msDS-AzOperationId attribute of these objects. The range of reserved values for this attribute is as follows: Command: (HEX) 0500,0000 – 05FF,FFFF PAM application: (HEX) 0200,0000 – 02FF,FFFF If this attribute is set to a value that is out of the reserved range, the right will be invalid and will no longer appear in Access Manager. If the right has been assigned to a role, the Analyze Invalid right assignment check returns an error. You can select the error in the **Analysis Results** node and use the Action menu to delete it from the role if you wish.
Invalid role assignments	Check whether invalid role assignments exist in the zone. In most cases, invalid role assignments occur when a role assignment is defined for a computer account and the computer leaves a zone without cleaning up roleassignment objects.
Invalid role	Check for role assignments that contain multiple roles or multiple users. In most cases, this error only occurs if you are using third-party tools to edit role assignments. Centrify tools prevent you from creating invalid role assignments. Note that a role assignment consists of a

assignments (DZ V2)	single user and a single role. To assign multiple roles to a user, you create multiple role assignments, which are stored in the form of user@domain role/sourceZone; for example: qa1@acme.com login/engineering qa1@acme.com vi_power/engineering qa1@acme.com test/engineering
Orphan child zones	Check for child zones that have an invalid parent zone. The information identifying the parent-child zone relationship is stored in the child zone in the form of a HEX string and the name of the domain to which the parent zone belongs. If this identifier is deleted, or changed to an invalid format, or if the parent zone is deleted but the child zone remains in the domain, Analyze (Orphan child zones) returns an error. Note that this error typically occurs only if you use third-party tools to edit zone objects in Active Directory. If you delete a parent zone using Centrify tools, child zones are deleted as well.
Orphan role assignments	Check for role assignments that consist of a non-existent role or user, or that do not contain a role or user. In most cases, this error only occurs if you are using third-party tools to edit Centrify objects in Active Directory. If you delete a role or user using Centrify tools or using Active Directory Users and Computers, the role assignment will be deleted as well (the change will be visible after you refresh the display) and Analyze will not return an error.
Orphan zone data objects and invalid data links	Check for zone data that have no corresponding Active Directory objects or have invalid links to Active Directory objects. For example, if you delete an Active Directory user but do not remove the profile for this user in a zone, the zone profile becomes an orphan and is flagged as such by this option.
Restricted roles	Check for roles that have been assigned commands that cannot be executed. When rights are created, they can be defined to run in a restricted-shell role, in an enhanced role (with dzdo), or with both. If a command that has not been defined to run in a restricted-shell is added to a restricted-shell role, this check returns an error.
Zone created under another zone	Check for zone information created in another zone's parent container. Note that this check does not look at hierarchical zones because it is expected that child zones are physically contained in their parent zone.
Zone information in old format	Check for zone information stored in an obsolete Centrify zone format.
Zoneless computers	Check for computers that do not belong to any zone.

4. Review the result summary, then click **Finish**.
5. If the result summary indicates any issues, you can view the details by selecting **Analysis Results** in the console tree and viewing the information listed in the right pane. For example:

   Analysis results

   For additional information, select the warning or error, right-click, then select **Properties**. For example:

   Properties

Common scenarios that generate analysis results

For most organizations, it is appropriate to check the data integrity of the Active Directory forest on a regular basis. Although running the Analyze command frequently may not be necessary for small networks with few domain controllers, there are several common scenarios that you should consider to determine how often you should check the forest for potential problems. The most likely reasons for data integrity issues stem from:

- Multiple administrators performing concurrent operations.
- Administrators using different domain controllers to perform a single operation.

- Replication delays that allow duplicate or conflicting information to be saved in Active Directory.
- Insufficient permissions that prevent an operation from being successfully completed.
- Network problems that prevent an operation from being successfully completed.
- Partial or incomplete upgrades that result in inconsistency of the information stored in Active Directory.
- Using ADEdit rather than the Console to create, modify, or delete zone objects, which may lead to problems, such as inadvertently creating a circular zone structure or an empty profile.
- Using third-party tools, such as ADSI Edit, to edit objects directly in Active Directory, which may lead to corrupted or invalid zone objects.

Running Analyze periodically helps to ensure the issues these scenarios can cause are reported in the Analysis Results, so you can take corrective action.

Responding to analysis results

Depending on the type of warning or error generated in the Analysis Results, you might be able to take corrective action or access additional information by right-clicking a result, then selecting an appropriate action. For example, if a computer account lacks the permission required to update Active Directory with the operating system version currently installed, you can right-click the warning in the Analysis Result then select **Grant computer the rights to modify operating system properties**.

If right-clicking a result does not provide a responsive action, you should use Access Manager or ADEdit to correct the issue.

The following table describes the warnings and errors you may see in the Analysis Results after running the Analyze wizard and how to resolve potential issues.

If there are any computers joined to multiple zones, an error is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if an administrator runs adleave with the --force option then runs adjoin to join the computer to a different domain without removing the old computer profile from Active Directory. You should identify the appropriate zone for the computer, then use the Access Manager console to delete the computer profile from any additional zones.
If the parent-child relationship of any zones is circular, an error is displayed.	Break the circular relationship.
If there are any duplicate groups in a zone, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate group profile to be added to a zone. For example, if two administrators add the same group to a zone using different domain controllers, there will be duplicate group profiles after the domain controllers complete replication. You should use the Access Manager console or ADSI Editor to delete the duplicate group profiles from the zone.
If any duplicate service principal names (SPNs) are found for users or computers in the forest, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. Right-click the warning and click **Properties** to identify the duplicate SPN. Open the account properties for the user or computer and modify or remove the duplicate servicePrincipalName value. Alternatively, run the adjoin command with the -d or --forceDeleteObjWithDupSpn option. See the adjoin man page for additional information.
If there are any duplicate users in a zone, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate user profile to be added to a zone. For example, if two administrators add the same user to a zone using different domain controllers, there will be duplicate user profiles after the domain controllers complete replication. You should use the Access Manager console or ADSI Editor to delete the duplicate user profiles from the zone.
If more than one Centrify SFU zone is found in the forest, a warning is	No responsive action can be taken directly within the Analysis Results for this issue. Because an SFU zone is associated with an Active Directory SFU schema extension, there should be a maximum of one SFU zone in an Active Directory forest. In general, this issue only occurs if multiple administrators perform concurrent

displayed.	operations or there are replication delays that allow a duplicate. You should use the Access Manager console or ADSI Editor to delete any duplicate SFU zones.
If a duplicate default parent container for zones is found, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate default container for new zones. Having more than one default parent container for zones can result in an unexpected default value in the Create New Zone wizard. You should use the ADSI Editor to delete any duplicate Zones parent containers from the forest.
If a computer role does not have any member computers or role assignments, a warning is displayed.	If the computer role has no member computers, right-click the warning in the Analysis Results, then select **Add computers** to add computers, or Delete Computer Role to remove the computer role. If a computer role has computer members but no role assignments, the only available response from the Analysis Results zone is to delete the computer role. You can, however, select the computer role in the Console, and add role assignments to its Role Assignments node.
If a user or group profile has been added to a zone but has no attributes defined, an error message is displayed.	Right-click the warning in the Analysis Results, then select **Delete empty profile** to delete the profile from the zone, or **Modify profile** to define one or more attributes for the user or group.
If any zone does not contain users, groups, or computers, a warning is displayed for each type of object. For example, if a zone has computers and groups, but no users, only the user warning is displayed for that zone.	No responsive action can be taken directly within the Analysis Results for these issues. In general, this issue occurs early in a deployment before you have populated zones. You should use the Access Manager console to add missing objects to the zone. If the empty zone is not a valid zone, right-click the zone and select **Delete**.
If one or more secondary profiles are found for a user but no primary profile is found, a warning message is displayed.	Right-click the warning in the Analysis Results, then select **Promote secondary profile to primary** to select a secondary profile you want to make the primary profile for the user.
If a user's UNIX profile is incomplete in the entire zone hierarchy, a warning message is displayed.	Right-click the warning in the Analysis Results, then select **Modify zone profile** to define additional attributes to complete the user's profile.
If the Active Directory group zone_nis_servers is not found in a zone configured for agentless authentication, an error is displayed.	Right-click the error in the Analysis Results, then select **Create NIS servers group** to create the zone_nis_servers group for agentless authentication. Note that your account must have permission to create this object for the operation to be successful.
If the membership of the zone_nis_servers group is not consistent with the computers authorized as NIS servers, a "Membership inconsistent" error is displayed.	Right-click the error in the Analysis Results, then select **Fix group membership** to modify the membership list for the zone_nis_servers group.
If a zone is configured to support agentless authentication and the zone_nis_servers group exists but does not contain all computers in the zone, an informational alert is displayed.	No responsive action can be taken directly within the Analysis Results for these issues. You should verify that all of the computers you want to use as NIS servers in the zone are configured to allow agentless authentication.
If there is a discrepancy between the DNS name in AD and the Centrify computer profile name, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Fix group membership** to

If a computer account does not have permission to write to the keywords attribute, an error is displayed.	Right-click the error in the Analysis Results, then select **Grant permission to computer account** to update the permissions on the computer account object.
If a computer account does not have permission to modify operating system properties, a warning is displayed.	Right-click the error in the Analysis Results, then select **Grant computer permission to modify operating system properties** to update the permissions on the computer account object.
If a right for a role is invalid, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Delete Right** to delete the right from the role.
If a role assignment is invalid, a warning message is displayed.	
If multiple roles are assigned to a user, a warning message is displayed.	
If a child zone has an invalid parent zone, an error message is displayed.	
If an object has no parent object, a warning message is displayed.	
If a restricted-shell role is assigned a right that cannot be run in a restricted shell, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Delete Commands** to remove the commands from the role, or select **Allow running in restricted role** to allow running the command in the restricted role.
If a zone was created using the version 2.x console and includes a Private Groups container, a warning is displayed.	If any computers in the zone are running version 2.x or 3.x agents, you should ignore this warning to ensure compatibility for those agents. If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove privateGroupCreation attribute** to update the zone format.
If a computer profile was created using the version 2.x console, the warning "Unix computer is in old format" is displayed.	If any computers in the zone are running version 2.x or 3.x agents, you should ignore this warning to ensure compatibility for those agents. If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy and unix_enabled attribute** to update the computer profile in the zone.
If a group profile was created using the version 2.x console, the warning "Unix group is in old format" is displayed.	If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy attribute** to update the group profile in the zone.
If a user profile was created using the version 2.x console, the warning "Unix user is in old format" is displayed.	If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy and app_enabled attribute** to update the user profile in the zone.
If a computer, group, or user profile exists, but no corresponding Active Directory computer, group, or user object is found, the warning "Orphan UNIX data object" is displayed.	In general, this issue occurs if an administrator removes an Active Directory computer, group, or user object manually using ADSI Editor or Active Directory Users and Computers but the corresponding data is not removed for the UNIX profile. Right-click the warning in the Analysis Results, then select **Remove orphan profile** to remove all of the UNIX properties associated with the orphan profile.
If a computer, group, or user profile has inconsistent links, an informational "Inconsistent links" alert is displayed.	Computer, group, and user profiles are associated with Active Directory computer, group, and user objects through either the managedBy attribute (agent version 2.x) or a parentLink value in the keywords attribute (agent version 3.x and later). If the links refer to different Active Directory objects, you will see this alert. Right-click the alert in the Analysis Results, then select **Overwrite with the active link** to remove outdated links.

If a computer, group, or user profile does not have a parentLink value defined, a “Missing parentLink” warning is displayed.	Right-click the warning in the Analysis Results, then select **Missing parentLink** to add the parentLink value to the keywords attribute.
If the parent container for a zone is another zone object, an error is displayed.	No responsive action can be taken directly within the Analysis Results for these issues. You should move the zone to another parent container or delete and recreate the zone in a different location.
The computer ObjectName contains Centrify information but it is not in a zone.	Right-click the warning in the Analysis Results, then select **Move to Zone** to search for and select the zone you want to place the computer in.

Configuring logging for the agent

By default, the Centrify Agent for *NIX logs errors, warnings and informational messages in the UNIX syslog and /var/log/messages files along with other kernel and program messages. Although these files contain valuable information for tracking system operations and troubleshooting issues, occasionally you may find it useful to activate agent-specific logging and record that information in a log file.

To enable logging on the Centrify Agent for *NIX

1. Log in as or switch to the root user.
2. Run the addebug command:

   /usr/share/centrifydc/bin/addebug on

   Note: You must type the full path to the command because addebug is not included in the path by default.

   Once you run this command, all of the Centrify Agent activity is written to the /var/log/centrifydc.logfile. If the adclient process stops running while you have logging enabled, the addebug program records messages from PAM and NSS requests in the /var/centrifydc/centrify_client.log file. Therefore, you should also check that file location if you enable logging.

For performance and security reasons, you should only enable logging when necessary, for example, when requested to do so by Centrify Support, and for short periods of time to diagnose a problem. Keep in mind that sensitive information may be written to this file and you should evaluate the contents of the file before giving others access to it.

When you are ready to stop logging activity, run the addebug off command.

Setting the logging level

You can define the level of detail written to the log by setting the log configuration parameter in the Centrify configuration file:

log: level

With this parameter, the log level works as a filter to define the type of information you are interested in and ensure that only the messages that meet the criteria are written to the log. For example, if you want to see warning and error messages but not informational messages, you can change the log level from INFO to WARN. By changing the log level, you can reduce the number of messages included in the log and record only messages that indicate a problem. Conversely, if you want to see more detail about system activity, you can change the log level to INFO or DEBUG to log information about operations that do not generate any warnings or errors.

You can use the following keywords to specify the type of information you want to record in the log file:

FATAL	Fatal error messages that indicate a system failure or other severe, critical event. In addition to being recorded in the system log, this type of message is typically written to the user's console. With this setting, only the most severe problems generate log file messages.
ERROR	System error messages for problems that may require operator intervention or from which system recovery is not likely. With this setting, both fatal and less-severe error events generate log file messages.
WARN	Warning messages that indicate an undesirable condition or describe a problem from which system recovery is likely. With this setting, warnings, errors, and fatal events generate log file messages.
INFO	Informational messages that describe operational status or provide event notification.

Logging for Access Manager

Although most logging activity focuses on the actions of the Centrify Agent, you can also enable or disable logging for the Access Manager console and configure the types of messages to record in the log file by selecting options in Access Manager.

To configure logging for operations handled through the Access Manager console:

1. Open Centrify Access Manager.
2. In the console tree, select **Centrify Access Manager**, right-click, then click **Options**.
3. Click the **Log Settings** tab, select the type of messages to log, then click **OK**.

If you enable logging, the log file is located by default in the C:\Users\user\AppData\Roaming\Centrify\DirectControl folder and is updated as you perform different operations in the Access Manager console.

Logging to the circular in-memory buffer

If the Centrify Agent for *NIX's adclient process is interrupted or stops unexpectedly, a separate watchdog process (cdcwatch) automatically enables an in-memory circular buffer that writes log messages passed to the logging subsystem to help identify what operation the adclient process was performing when the problem occurred. The in-memory buffer is also mapped to an actual file, so that if there's a system crash or a core dump, the last messages leading up to the event are saved. Messages from the in-memory circular buffer have the prefix _cbuf, so they can be extracted from a core file using the strings command.

The in-memory circular buffer allows debug-level information to be automatically written to a log file even if debugging is turned off. It can be manually enabled by restarting the adclient process with the -M command line option. The default size of the buffer is 128K, which should be sufficient to log approximately 500 messages. Because enabling the buffer can impact performance, you should not manually enable the circular buffer or modify its size or logging level unless you are instructed to make the changes by Centrify Support.

Collecting diagnostic information

You can use the adinfo command to display or collect detailed diagnostic and configuration information for a local UNIX computer. Options control the type of information and level of detail displayed or collected. The options you are most likely to use to collect diagnostic information are the --config, --diag, or --support options, which require you to be logged in as root. You can redirect the output from any adinfo command to a file for further analysis or to forward information to Centrify Support.

For more information about the options available and the information returned with each option, see the man page for adinfo.

To display the basic configuration information for the local UNIX computer, you can type:

adinfo

If the computer has joined a domain, this command displays information similar to the following:

```
Local host name: magnolia
Joined to domain: ajax.org
Joined as: magnolia.ajax.org
Current DC: ginger.ajax.org
Preferred site: Default-First-Site-Name
Zone: ajax.org/Ajax/Zones/corporate
Last password set: 2006-12-28 14:47:57 PST
CentrifyDC mode: connected
Licensed Features Enabled
```

Working with domain controllers and DNS servers

Delinea Agents are designed to perform the same set of DNS lookup requests that a typical Windows workstation performs to find the nearest domain controller for the local site. The DNS lookup request enables the Centrify Agent for *NIX to find domain controllers as they become available on the network or as the computer is relocated to another network location where different domain controllers are present. Centrify Agents also use DNS to find the Kerberos service providers and the global catalog service providers for the Active Directory forest.

In a typical Windows environment, the DNS server role is updated dynamically to contain the service locater (SRV) DNS entries for Active Directory's LDAP, Kerberos, and global catalog services, so this information is available for Centrify Agents to use. However, there are some configurations of DNS that might not provide all of the SRV records for the set of domain controllers that provide Active Directory service to the enterprise. You may also run into problems if DNS for the enterprise runs on UNIX servers that cannot locate your Active Directory domain controllers. The next sections describe how you can adjust DNS or Centrify Agent to ensure they work together properly in your environment.

Related topics

Configuring the DNS server role on Windows

One of the most common scenarios for running DNS in an environment with Active Directory is to add the DNS server role to a Windows domain controller or another Windows server.

If you are already using DNS in Active Directory and dynamically publishing DNS service records, no additional configuration should be necessary. If you are using DNS in Active Directory but have disabled dynamic updates, you should change the configuration for the DNS server role to allow dynamic updates. Making this change will allow Centrify Agents to properly locate domain controllers in the site and select an appropriate new domain controller if a connection to its primary domain controller is lost or the managed computer is moved to a new location on the network.

Configuring DNS running on UNIX servers

If your environment is configured to use UNIX-based DNS servers instead of Active Directory-based DNS servers and the UNIX system is configured to use DHCP, the nameserver entry in /etc/resolv.conf file is set automatically to point to a DNS server.

If this DNS server is aware of the Active Directory domain you want to join, no further changes are needed. If the DNS server identified as a nameserver in the /etc/resolv.conf file is not aware of the domain you are trying to join, for example, because you are using a test domain or a separate evaluation environment, you need to either disable DHCP or manually set the location of the Active Directory domain controller in the Centrify configuration file.

Checking whether DNS can resolve the domain controller

In most cases, you can verify whether a UNIX computer can locate the domain controller and related services by running the ping command and verifying connectivity to the correct Active Directory domain controller or by checking the nameserver entry in the /etc/resolv.conf file. This nameserver entry should be the IP address of one of the domain controllers in the domain you want to join.

If the ping command is successful, it indicates the DNS server is aware of the Active Directory domain you want to join and no further changes are needed. If the ping command is not successful, you will need to take further action to resolve the issue.

Resolving issues in locating Active Directory domain controllers

If the UNIX computer cannot find the Active Directory domain controller, there are several ways you can resolve the issue. Depending on your environment and specific situation, you should consider doing one of the following:

- Set up DNS on the target Active Directory domain controller and the manually configure the nameserver entry in the /etc/resolv.conf file to use that domain controller as described in Setting up DNS service on a target domain controller.
- Set the Centrify configuration file to manually identify the domain controllers you want to use as described in Setting the domain controller in the configuration file.

Setting up DNS service on a target domain controller

One of the simplest ways to ensure that the UNIX computers can locate the Active Directory domain controller and related services is to use the DNS service on the Active Directory domain controller as a DNS slave to the enterprise DNS servers. You can do this is by configuring the DNS server role on the Active Directory domain controller, then specifying that domain controller in the UNIX computer's /etc/resolv.conf file. You can then add a forwarder to the local DNS on the domain controller that will pass on all lookups that it cannot satisfy to an enterprise DNS server.

This configuration does not require any changes to the enterprise DNS servers. Any look up request from the domain controller is simply a query from another computer in the enterprise. However, the UNIX computers configured to use this slave DNS service will receive the appropriate Service Location (SRV) records and global catalog updates for the Active Directory domain controller. In addition, the DNS service on the domain controller can be configured to forward requests to the enterprise DNS servers so those requests can be answered when the local DNS service cannot respond.

Adding a DNS server role to an Active Directory domain controller

The specific steps for adding the DNS server role to a domain controller depend on the version of Windows Server you use. In most cases, you can use an administrative tool, such as Server Manager, to add roles. Follow the instructions displayed in the wizard to add the **DNS Server** server roles, configure the DNS server lookup zones, select the **Allow both nonsecure and secure dynamic updates** option.

After you have configured the DNS server role on the domain controller, the computer uses the local DNS server as its primary DNS server.

Configuring UNIX to use DNS service on the target domain controller

Once you have configured the DNS service to contain the required Active Directory entries, you simply need to modify the UNIX computer to send all DNS lookup requests to the newly configured DNS server.

To configure the UNIX computer to use the new DNS server:

1. Open the /etc/resolv.conf file.
2. Set the IP address of the nameserver entry to the IP address of the DNS server on the Active Directory domain controller you just configured.

Setting the domain controller in the configuration file

If you are not able to use DNS to locate the Active Directory domain controllers on your network, you can manually specify one or more domain controllers in the Centrify configuration file.

To manually specify a domain controller, add the following entry to the Centrify configuration file, /etc/centrifydc/centrifydc.conf:

dns.dc.domain_name: server_name [server_name ...]

For example, if you want to ensure the Centrify Agent uses the domain mylab.test and the domain controller named dc1.mylab.test, you could add the following line to the /etc/centrifydc/centrifydc.conf file:

dns.dc.mylab.test: dc1.mylab.test

Note: You must specify the name of the domain controller, not its IP address. In addition, the domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify multiple servers for a domain, use a space to separate the domain controller server names. For example:

dns.dc.mylab.test: dc1.mylab.test dc2.mylab.test

The Centrify Agent will attempt to connect to the domain controllers in the order specified. For example, if the domain controller dc1.mylab.test cannot be reached, the agent will then attempt to connect to dc2.mylab.test.

If the global catalog for a given domain is on a different domain controller, you can add a separate dns.gc.domain_name entry to the configuration file to specify the location of the global catalog. For example:

dns.gc.mylab.test: dc3.mylab.test

You can add as many domain and domain controller entries to the Centrify configuration file as you need. Because the entries manually specified in the configuration file override any site settings for your domain, you can completely control the Centrify Agent for *NIX's binding to the domains in your forest through this mechanism.

Note: In most cases, you should use DNS whenever possible to locate your domain controllers. Using DNS ensures that any changes to the domain topology are handled automatically through the DNS lookups. The settings in the configuration file provide a manual alternative to looking up information through DNS for those cases when using DNS is not possible. If you use the manually-defined entries in the configuration file and the domain topology is changed by an Active Directory administrator, you must manually update the location of the domains in each configuration file.

Using the fixdns script

The Centrify Agent includes a fixdns script that you can use to inspect your environment and make the necessary configuration file changes for you.

To run this script, you need to specify the domain controller name and IP address:

fixdns domain_controller_name IP_address

For example if you intend to join the domain mytest.lab and the domain controller for that domain is dc1.mytest.lab and its address is 172.27.20.1, you would run the following command:

fixdns dc1.mytest.lab 127.27.20.1

The fixdns script will then make the necessary changes to the /etc/hosts and the Centrify configuration file.

Note: This script does not update the /etc/resolv.conf file. If the script cannot locate the domain controller using the existing /etc/resolv.conf settings, it will assume that you want to use settings from the configuration file.

What the Centrify DNS subsystem provides

Centrify provides a DNS subsystem that bypasses the local DNS resolver to address common issues that occur with many local DNS resolvers. These common issues for local DNS resolvers include:

- Degraded performance when connecting to a slow DNS server or when attempting to use dead DNS servers.
- Degraded performance when reacquiring a DNS server that went offline and has come back online.
- Degraded performance related to DNS timeouts.
- Platform-related DNS idiosyncrasies, such as MDNS, appending.LOCAL suffixes, and so on.

The Centrify DNS subsystem performs the following functions:

- Looks up hosts by name.
- Looks up hosts by IP address.
- Queries DNS service location records (SRV) to discover the domain controllers that support Active Directory services including KDC, KPASSWD, LDAP and the global catalog.

Resolving a host name or IP address

When the DNS client subsystem receives a DNS requests, it attempts to resolve the host name or IP address by first checking the /etc/hosts file. If the file contains a valid entry to resolve the specified host name or IP address, the DNS client subsystem processes the DNS request.

Entries in /etc/hosts must be in the following format:

IPv4_address hostname alias alias ...

where:

- IPv4_address must be in the first position
- hostname is a fully-qualified domain name and must be in the second position.
- aliases are optional and follow the address and hostname entries.

For example:

192.169.147.135 ginger.acme.com ginger

Note: Service (SRV) record queries cannot be satisfied from the /etc/hosts file.

If resolution by /etc/hosts is unsuccessful, the DNS subsystem attempts to select a DNS server that can be used to resolve the host name or IP address (as described in the next section, Selecting a DNS server).

Selecting a DNS server

If unable to resolve a host name or IP address by finding an entry in the /etc/hosts name (as described in Resolving a host name or IP address), the Centrify DNS subsystem attempts to find a DNS server to resolve the host name or IP address, as follows:

- It checks for a working DNS server that has already been selected (cached in memory and stored in /var/centrify/kset.dns.server), and if available, uses it.
- If a working DNS server is not already selected, it checks /etc/resolv.conf for configured DNS servers, and if populated, selects the fastest one from the list.

If no working DNS servers are found, the request fails.

At this point, DNS is considered down, and the Centrify DNS subsystem waits for the interval specified by the dns.dead.resweep.interval (default is 60 seconds), before attempting again to find a DNS server.

Specifying DNS-related parameters

Parameters in the Centrify configuration file control many aspects of Centrify DNS subsystem operation. Although you can set any of these parameters, the default settings should provide you with optimal DNS operation. See the Configuration and Tuning Reference Guide for details about any of these parameters.

The DNS subsystem periodically checks in the background to see if a DNS server that is faster than the currently selected one is available. The dns.alive.resweep.interval parameter determines how often this background check occurs; the default value is one hour (3600 seconds).

When a DNS server is selected, its address is stored in the kset.dns.server file, and it is used for all DNS requests until one of the following occurs:

- The selected server stops responding.
- A new server sweep discovers a faster DNS server and replaces it.
- The adclient process is stopped and restarted, which triggers a sweep for a new DNS server.
- The specified server is no longer in the list of servers in /etc/resolv.conf.

For the sweep, the dns.sweep.pattern parameter determines the probe pattern that is used to find a live DNS server; that is, it sets the protocol to use (TCP or UDP) and the amount of time to wait for a response. By default, this parameter specifies both a TCP and UDP probe.

The dns.timeout and dns.udp.retries parameters determine the amount of time to wait, and how often to re-send a request when the current server does not respond to a request. If the current server does not respond to a request within the specified time out period, it is considered down and Centrify looks for a different server. If it cannot find a live server, DNS is considered down, and the Centrify Agent for *NIX waits for the period of the dns.dead.resweep.interval parameter, 60 seconds by default, before performing a sweep to find a new server.

Filtering the objects displayed

For performance or security reasons, you might want to filter or limit the objects displayed in the Access Manager console. Depending on your environment, you might want to display more or less information by setting filter options. These filter settings enable you to control both the number and type of objects displayed. You should note, however, that these settings can affect the performance of the console.

To filter the objects listed in Access Manager:

1. Open Access Manager.
2. In the console tree, select **Access Manager**, right-click, then click **Options**.
3. Click the **Filter Settings** tab.
4. Select **Load all zones** to automatically open either all zones in the connected forest or all zones in a specific parent container.
    - If you select this option and **connected forest** all zones in the forest are opened automatically each time you start Access Manager. Selecting this option prevents you from opening or closing any zones manually. Depending on the number of zones you have, you might experience slower performance in the console if you select this option.
    - If you select this option and **container**, you can then click Browse to search for a container from which to automatically load zones. Selecting this option prevents you from opening or closing any zones manually. Depending on the number of zones you have in the selected container, you might experience slower performance in the console if you select this option.
5. Select **Show disabled Active Directory accounts** to display disabled computer and user accounts or uncheck this option to hide disabled objects.
6. Select **Show orphans** to display all users, groups, and computers that have a UNIX profile or uncheck this option to hide all orphan profiles.

   Orphan profiles are the service connection points that no longer have a corresponding Active Directory object. Hiding or removing orphan profiles can improve console performance. For information about locating orphan profiles by running an analysis on the Active Directory forest, see Analyzing information in Active Directory.
7. Select **Show Auto Zone** to display the users, groups, and computers that have joined the Auto Zone or uncheck this option to hide Auto Zone information.
8. Set the **Maximum number of items to be displayed in the list** option to limit the total number of objects displayed in the console, up to total maximum allowed (65535).

   This setting applies to all of the objects displayed in Access Manager, including zones, computers, users, groups, pending users, pending groups, NIS maps, and all defined rights, roles, and role assignments. Lowering the maximum number of items displayed improves performance when browsing the listed items. Note that this setting does not affect the number of items you can define, only the number displayed.
9. Click **OK**.

Authentication Service issues on

Be aware of the following issue when working with the Authentication Service on UNIX or Linux systems:

- The directory /var should not be NFS mounted or else DirectControl may not work properly. (Ref: IN-90009)
- Please see KB-9092 for further details about using common UNIX commands with Privilege Elevation Service restricted shells.

Using Centrify commands for administrative tasks

This chapter provides an overview of the Centrify command-line interface and a list of the command-line programs you can execute locally on Centrify-managed computers.

How and when to use command-line programs

UNIX command-line programs are installed by default when you install the Centrify Agent for *NIX. The commands are typically installed in one of the following directories:

/usr/sbin, /usr/bin
/usr/share/centrifydc /bin

The Centrify Agent includes a large number of command-line programs that enable you to perform a variety of administrative tasks directly from a UNIX shell or using a shell script. These command-line programs use the underlying adclient service library to perform important tasks on the computers you add to Active Directory domains. For example, there are commands that allow you to remove a computer from an Active Directory domain, change an Active Directory user's password, and return detailed diagnostic information about the operations of a host computer.

You can use command-line programs interactively or in shell scripts when you must take action directly on a Centrify-managed computer, or when taking action from a managed computer is most convenient. For example, individual users can use a command-line program to change their Active Directory password from a login shell without logging on to a Windows computer.

Some command-line programs perform specific tasks that you will only use infrequently or under specific conditions. Other programs perform common administrative tasks that you are likely to use repeatedly.

The most commonly used programs include the following:

- The adjoin command is the first command you use to add a local computer to an Active Directory domain.
- The adinfo command display summary or detailed diagnostic and configuration information for a computer and its Active Directory domain.
- The adpasswd command allows you to change an Active Directory account password from a Centrify-managed computer.
- The adgpupdate command allows you to force group policies to be refreshed immediately.
- The adleave command allows you to remove a managed computer from its current Active Directory domain or from the Active Directory forest entirely.

Displaying usage information and man pages

To display a summary of usage information for any command-line program, type the command and the --help or -h option. For example, to see usage information for the adleave command, type:

adleave --help

The usage information includes a list of options and arguments, and a brief description of each option. For example, if you specify adleave -h on the command line, the command displays the command-line syntax and a list of the valid options you can use when you execute adleave commands, similar to the following:

usage: adleave [options]
options:
-u, --user user[@domain] user name, default is administrator
-p, --password pw user password, prompts if absent
-s, --server ds domain server for leave operations
-Z, --zoneserver ds domain server for zone operations
useful if zone is in another domain
-C, --noconf do not restore PAM or NSS config
-G, --nogp do not restore Group Policy
-f, --force force local leave, no network activity
-v, --version print version information
-h, --help print this help information and exit

For more complete information about any command, you can review the information in the command's manual (man) page. For example, to see the manual page for the adleave command, type:

man adleave

Result codes used by multiple programs

Many Centrify command-line programs share a common set of result codes returned when an operation is successful or an error occurs. The following table lists the result codes that are reserved for use by Centrify command-line programs.

0	ERR_SUCCESS	Successful completion of the operation.
6	ERR_OTHERS	Miscellaneous errors occurred during the operation.
7	ERR_USAGES	Usage error occurred during the operation.
8	ERR_OP_ABORTED	Operation aborted by user.
9	ERR_ROOT_PRIV	Root privilege is required for the operation.
10	ERR_NOT_JOINED	Computer is not currently joined to any Active Directory domain.
11	ERR_ALREADY_JOINED	Computer is already joined to the current Active Directory domain.
12	ERR_JOINED_ANOTHER_DOMAIN	Computer is currently joined to another Active Directory domain.
13	ERR_ADCLIENT_DOWN	The adclient process is not running or not available.
14	ERR_ADCLIENT_DISCONNECTED	The adclient process is running in disconnected mode.
15	ERR_ADLCIENT_STARTUP	The adclient process failed to start.
16	ERR_DNS_TIMEOUT	The DNS server is not responding and may be down.
17	ERR_DNS_GENERIC	A generic DNS problem occurred during the operation.
18	ERR_INVALID_DOMAIN_NAME	The Active Directory domain name is incorrect or not found in DNS.
19	ERR_INVALID_LOGON	User name or password provided is not correct.
20	ERR_ACCOUNT_DISABLED	The account specified has been disabled.
21	ERR_ACCOUNT_EXPIRED	The account specified has expired.
22	ERR_ACCOUNT_EXISTS	The account specified already exists,
23	ERR_ACCOUNT_NOTFOUND	The account specified was not found in Active Directory.
24	ERR_PASSWORD_EXPIRED	The account password has expired.
25	ERR_ZONE_NOTFOUND	The zone cannot be found.
26	ERR_CONTAINER_NOTFOUND	Invalid Active Directory container object.
27	ERR_INSUFFICIENT_PERM	The account specified does not have permission to perform the operation.
28	ERR_CLOCK_SKEW	The time difference between system clocks is beyond the acceptable range.
29	ERR_COMPUTER_NAME	Invalid computer account.
30	ERR_CRED_INVALID	Invalid credentials.

31	ERR_SERVICE_TKT_INVALID	Invalid service ticket.
32	ERR_POLICY_NOT_MATCH	Policy not matched.
33	ERR_REJECT_CHG_PASSWD	Password change rejected.
34	ERR_WORKSTATION_DENY	Workstation denied.
35	ERR_NOT_FIND_USER	No matching user found.
36	ERR_NOT_FIND_GROUP	No matching group found.
37	ERR_NOT_CONNECT_ADCLIENT	An attempt to open a connection to the adclient process failed.
38	ERR_ADLCIENT_STOP	Unable to stop the adclient process.
39	ERR_QUOTA_EXCEEDED	The user has exceeded the number of join operations allowed.
40	ERR_OPEN_FILE	The attempt to open a file failed.
41	ERR_READ_FILE	The attempt to read a file failed.
42	ERR_COPY_FILE	The attempt to copy a file failed.

For information about command-specific result codes, see the manual page for individual commandline programs.

Delinea

Perform administrative tasks using commands

Most administrative tasks can be performed using Access Manager on a Windows computer or by using ADEdit commands or scripts from a Centrify-managed computer that has access to the Active Directory domain controller. In some cases, however, there are operations that you must or prefer to perform locally on a managed computer by executing commandline programs.

The command line programs allow you to perform administrative tasks—such as join or leave a domain or generate diagnostic information—directly in a UNIX shell. Many of the command-line programs require administrative privileges or must run using root to perform privileged operations. You can define command rights for these programs to grant permission to run them to other users.

The following table provides a summary of the command-line programs for access control and privilege management that are installed with the Centrify Agent for *NIX. For complete information about the options you can specify for any command, see the man page for that command.

adcache	Clear the local cache on a computer. You can use this command to clear all cached information or a specific cache file. You can also use the command to check a cache file for a specific key value and to reclaim disk space.
adcheck	Check the operating system, network, and Active Directory connections to verify that a computer is ready to join an Active Directory domain.
adchzone	Move a joined computer from a classic zone to a hierarchical zone. Before moving a computer with this command, you must use admigrate to migrate the classic zone to a hierarchical zone.
adclient	Start, stop, or manage operations for the Centrify Agent process on a local computer. In most cases, you should start and stop adclient using a startup script.
addebug	Start or stop detailed logging activity for the Centrify Agent (adclient) process on a local computer. If you do not specify an option, the addebug command displays its current status, indicating whether logging is active or disabled. You must be logged in as root to run this command.
addbloader	Create a database file with zone information. You can then use the adreport command to generate reports from this file, or read it with standard tools.
addns	Update DNS records on an Active Directory-based DNS server in environments where the DHCP server cannot update DNS records automatically.
adfinddomain	Display the domain controller associated with the Active Directory domain you specify.
adfips	Enable or disable FIPS-compliant encryption. You must be logged in as root to run this command.
adfixid	Resolve UID and GID conflicts and change the ownership of a local user's files to match the user and group IDs defined for the user in Active Directory.
adflush	Clear the cache on a local computer. Executing adflush with no options expires the domain controller and global catalog caches.
adgpupdate	Retrieve group policies from the Active Directory domain controller and apply the policy settings to the local computer and current user immediately.
adid	Display the real and effective UIDs and GIDs for the current user or a specified user.
adinfo	Display detailed Active Directory, network, and diagnostic information for a local computer. Options control the type of information and level of detail displayed.
adjoin	Add the local host computer to the specified Active Directory domain. You must log in as root to run the adjoin command.
adkeytab	Create and manage Kerberos key tables (*.keytab files) and coordinate changes with the Kerberos key distribution center (KDC) provided by Active Directory. The arguments required and options available depend on the operation you want to perform.

adleave	Remove the local host computer from its current Active Directory domain. You must log in as root to run the adleave command.
adlicense	Enable or disable licensed features on a local computer. You must log in as root to run the adlicense command.
admanagelocal	Display currently managed local accounts, status of local account management, and force a foreground sync of local accounts.
admigrate	Migrate information from a classic zone to a hierarchical zone. You can migrate a classic zone to a new peer hierarchical zone, or you can specify a parent zone for the migration.
adobfuscate	Obscure sensitive information, such as email addresses, host names, and user names, that might be recorded in a log file before sending the file to Centrify for analysis. You must create a pattern file to use with this command. The command reads the pattern file and replaces items matching the patterns specified with generic values.
adpasswd	Change the password of the user executing the command or change the password of another Active Directory user.
adquery	Query Active Directory for information about users and groups from the command line on a Centrify-managed computer. This command is provided for backward compatibility. In most cases, you should use adedit commands or scripts to perform administrative tasks in Active Directory from Linux or UNIX computers.
adreload	Force the Centrify Agent process (adclient) to reload the configuration properties in the /etc/centrifydc.conf file and in other files in the /etc/centrifydc directory.
adreport	Generate user, computer, command, and role assignment reports for a zone. You must run the addbloader command to create a database containing information about a zone before you can run this command to generate a report.
adrmlocal	Report and remove local user names that duplicate Active Directory user names.
adsendaudittrailevent	Specify where to send audit trail events. You can choose to send audit trail events to the syslog facility, the Centrify auditing service, or both.
adsetgroups	View or change the list of groups available for the current user.
adsmb	Perform file operations, such as get a file, write a file, or display the contents of a directory using the Centrify smb stack.
adupdate	Update user and group account information from the command line on Centrifymanaged computer. This command is provided for backward compatibility. In most cases, you should use adedit commands or scripts to perform administrative tasks in Active Directory from Linux or UJNIX computers.
dzdo	Execute a privileged command as root or another specified user. You must be assigned a role that grants privileged command rights to use this command.
dzedit	Edit a file as root or another user.
dzinfo	Display detailed information about the configuration of rights and roles for one or more specified users on the local computer. If you do not specify a user, the command returns information for the currently logged on user.
dzsh	Run commands in a restricted environment shell. This shell is a customized Bourne shell that provides environment variables, job control, command history, and access to specific commands defined by roles.
ldapadd	Open a connection to the Active Directory domain controller or another LDAP server to add new entries.
ldapcompare	Open a connection to the specified Active Directory domain controller or another LDAP server to compare LDAP entries. You can use this command to determine whether a specified entry has a particular attribute-value combination. The only information returned is whether the comparison evaluated to true or false. No other information about the entry is provided.

ldapdelete	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to delete the specified entry or entries.
ldapmodify	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to modify the specified entry or entries.
ldapmodrdn	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to move or rename the specified entry or entries.
ldapsearch	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to locate and retrieve the specified entry or entries.
nisflush	Clear the Centrify Network Information Service cache on a local computer, or restart the service without flushing the cache. You must be logged in as the root user to run this command.

Using Python with Centrify objects

You can use Python commands or scripts to interact with Centrify objects on Linux systems. Centrify provides two Python modules that you can use in Python scripts to communicate with the Centrify Agent for *NIX:

- **pylrpc**: Calls reference internal LRPC objects and methods.
- **pycapi**: Calls reference the Centrify API (CAPI) mainly for use with NSS and PAM.

Requirements

Python requirements:

- Use Python version 3.4 or later
- Add the following path to your PYTHONPATH variable:

  /usr/share/centrifydc/python/lib64

  For example, you can update the PYTHONPATH by running the following command:

  $export PYTHONPATH=/usr/share/centrifydc/python/lib64

In your python scripts, you can import either module into your script with an import statement, such as the following:

import pylrpc

import pycapi

There are some sample scripts included so that you can what kinds of things you can do. Example python scripts are installed at /usr/share/centrifydc/samples/python.

Python Pylrpc reference

This section covers the objects, methods, and other details for the Pylrpc module.

Pylrpc module objects

There are two objects in the Pylrpc module:

- Session

  This object works with the agent. When you construct this object, it creates a session with the agent automatically. When you delete this object, the session closes automatically.

- Error

  This is the type of exceptions that the Session object methods raise upon failure.

Pylrpc session object methods

This section lists out each method that you can use with the session object in the Pylrpc module.

__init__()

Opens a session with the agent.

adinfo()

Get joining settings and status of the local system

Parameters:

none

Returns:

A Python dictionary with keys and values that use the string type.

Raises:

- Error - if any error occurred

Example:

getUser(uid, option) and getUser(uname, option)

Query a user by UNIX UID, UNIX name or AD name

Parameters:

- uid (int) or name (str)
- option (int)
  - pylrpc.UNIX_ONLY : to ask adclient to return result only when the user is zone enabled

- pylrpc.CHECK_AD_FIRST: to ask adclient to ignore cache and read from AD if connected
- pylrpc.GROUP_MEMBERSHIP: to ask adclient to return user's group membership info
- pylrpc.EXPIRED_GRP_MEMBERS: when used with pylrpc.GROUP_MEMBERSHIP, ask adclient to trigger asynchronous group membership refresh for this user

Returns:

- Object (see Description of object below)

Raises:

- Error - if any error occurred

Example:

```
user = s.getUser("username", pylrpc.UNIX_ONLY)
user = s.getUser(999999, pylrpc.UNIX_ONLY | pylrpc.GROUP_MEMBERSHIP)

# Query an AD user by AD name

# by UPN or samAccountName@domain
user = s.getUser("Krusty@domain.com", pylrpc.GROUP_MEMBERSHIP)
# by NTLM name
user = s.getUser("domain.com+krusty", pylrpc.GROUP_MEMBERSHIP | pylrpc.CHECK_AD_FIRST)
# by Canonical name
user = s.getUser("domain.com/Users/krusty")
```

getGroup(gid, option) and getGroup(gname, option)

Query a zone group by gid or name

Parameters:

- gid (int) or name (str)
- option (int)
  - pylrpc.UNIX_ONLY : to ask adclient to return result only when the group is zone enabled
  - pylrpc.CHECK_AD_FIRST: to ask adclient to ignore cache and read from AD if connected
  - pylrpc.GROUP_MEMBERSHIP: to ask adclient to return group's group member info
  - pylrpc.EXPIRED_GRP_MEMBERS: when used with pylrpc.GROUP_MEMBERSHIP, ask adclient to trigger asynchronous member refresh for this group

Returns:

- Object (see Description of object below)

Raises:

- Error - if any error occurred

Example:

```
group = s.getGroup("username", pylrpc.UNIX_ONLY)
group = s.getGroup(999999, pylrpc.UNIX_ONLY | pylrpc.GROUP_MEMBERSHIP)

# Query an AD group by AD name

# by samAccountName@domain
group = s.getGroup("dba@domain.com", pylrpc.GROUP_MEMBERSHIP)
# by Canonical name
group = s.getGroup("domain.com/Users/dba")
```

flushCache(type)

Expire or flush adclient's cache

Parameters:

- type (int)
  - pylrpc.EXPIRE_OBJ_CACHE: force expire object data caches, equivalent to "adflush -e -fy"
  - pylrpc.FLUSH_DNS_CACHE: flush DNS cache, equivalent to "adflush -d -fy"
  - pylrpc.FLUSH_AUTH_STORE: flush authorization data cache, equivalent to "adflush -a -fy"
  - pylrpc.FLUSH_TRUSTS: flush domain trust cache, equivalent to "adflush -t -fy"
  - pylrpc.FLUSH_OBJ_CACHE: flush object data caches, equivalent to "adflush -o -fy"
  - pylrpc.FLUSH_BINDINGS: drop DC bindings, equivalent to "adflush -b -fy"
  - pylrpc.FLUSH_CONNECTORS: flush Centrify Connector info, equivalent to "adflush -c -fy"

Returns:

- True on success

Raises:

- Error - if any error occurred

Example:

refreshObject

force flush a single object out from object data cache

Parameters:

- type (int)
  - pylrpc.UserType
  - pylrpc.GroupType
- name (str)
  - Can be UNIX name or AD name

Returns:

- True on success

Raises:

- Error - if any error occurred

Example:

```
result = s.refreshObject(pylrpc.GroupType, "groupname")
```

Pylrpc Error object methods

The base class of Error is the Python Exception class.

Here's an example:

```
s = pylrpc.Session()

except pylrpc.Error as ex:
print("ERROR: %s, code= %s" % (ex.message(), ex.code()))
```

message()

The error message

Returns:

- message as (str)

code()

Returns the error code.

Returns:

- code as (int) (See codes and error messages)

Codes and error messages

9	Root privilege is required for the operation
10	The system is not joined to any domain
13	adclient is not running/not available
52	User not found in zone
35	Active Directory user not found
53	Group not found in zone
36	Active Directory group not found
6	Other misc errors

Pylrpc dictionary objects

Some of the pylrpc methods return objects, those are described below. A dictionary is a data type in Python that's used to store a set of key:value pairs.

Object	The Object is a dictionary object that stores the attributes of the object returned. For each item in the dictionary object, the key is a string, and the value is a list of bytes objects. If the attribute has only one value, the attribute will be a list with only one bytes object.

Python Pycapi Reference

This section covers the objects, methods, and other details for the Pycapi module.

Pycapi Module Methods

The following table provides a summary of the available methods in the pycapi module. Click the method name to go to the details for that method.

GetMajorVersion()	Returns the CAPI library's major version number.	int	The CAPI library's major version number
GetMinorVersion()	Returns the CAPI library's minorversion number.	int	The CAPI library's minor version number
Shutdown()	Does housekeeping in preparation for exiting a program that is using the CAPI library. Calling this function is optional, but if if the in-memory SID cache is enabled it will take care of freeing up any allocated memory associated with the cache.	n/a	
GetCdcCodeStr(code)	Returns the string associated with the supplied code. parameter: code (int) -code	string	The string associated with the code.
GetErrSystemStr(system)	Returns the name of the error subsystem with an ID. parameters: system (int) - error system ID	string	The name of the error subsystem.
DomainFromDN()	Returns the Active Directory domain name from the distinguished name or canonical name in upper case. Parameters: dn (string) - error system ID	string	The Active Directory domain name

Pycapi Module Objects

There are two objects in the Pycapi module:

- Session

  This object works with the agent. When you construct this object, it creates a session with the agent automatically. When you delete this object, the session closes automatically.

- Error

Session Object Methods

This section lists details about each method that you can use with the Session object.

–

Create a session with the agent using the open method.

–

Disconnect from the agent using the close method.

close()

Disconnect from the agent and free all resources associated with the session.

open(majorVersion, minorVersion)

Parameters:

- majorVersion(int): major version of required CAPI version
- minorVersion(int): minor version of required CAPI version

If you specify majorVersion:

- You must specify the major version of the Centrify API (CAPI). If the current version of CAPI is lower than the specified version, this method call fails.
- Optionally you can also specify the minorVersion.

If you don't specify the version parameters, the service doesn't do any version checking.

Raises

- Error - if any error occurred

getOption(option)

Get an option's current setting with an ID.

Parameters:

- option (int) - option ID (see Option constants)

Returns:

- value as (int)

Raises:

- Error - if any error occurred

setOption(option, value)

Set an option with an ID and a value.

Parameters:

- option (int): option ID (see Option in Constants)
- value (int): option value

Raises:

- Error - if any error occurred

setSessionID(id)

Set a session-specific string. This string will show up in the agent event logs to provide an easy way to track logging events specific to requests generated by this CAPI session.

Parameters:

- id (str) - session-specific string

Raises:

- Error - if any error occurred

isSessionConnected()

Check whether the session is connected to the DirectControl agent and the session is valid.

Returns:

- code as (int). If the session is connected and valid, the code value will be CODE_SUCCESS (see Code constants).

getSessionCode()

Get the code from the last session transaction.

Returns:

- code as (int) (see Code constants)

ldapFetch(domain, dn, attrs)

Fetch a specific object from Active Directory.

Parameters:

- domain (str) - domain to search in. Specify either a domain name, or "$" to use global catalog or "" to use the default domain controller.
- dn (str) - the DN to return. An empty string "" can be used to specify the DSE root.
- attrs (list of str) - the attributes to return. An empty list or None will return only the attributes DirectControl normally caches for the matched object.

Returns:

Object (see Object)

Raises:

Error - if any error occurred

lookupObjectByUnixId(type, id)

Look up a user or group by Unix ID.

Parameters:

- type (int) - object type (see Object type constants)
- id (int) - Unix user ID or group ID

Returns:

- Object (see Object)

Raises:

- Error - if any error occurred

lookupObjectByName(category, name)

Look up a user or group by name in a category.

Parameters:

- category (str) - category (see AD Category constants) to limit the search
- name (str) - user name or group name

Returns:

- Object (see Object)

Raises:

- Error - if any error occurred

lookupObjectByGuid(guid)

Look up a user or group by GUID.

Parameters:

- guid (str) - GUID

Returns:

- Object (see Object)

Raises:

- Error - if any error occurred

lookupObjectBySid(sid)

Look up a user or group by SID.

Parameters:

- sid (str) - SID

Returns:

- Object (see Object)

Raises:

- Error - if any error occurred

getDomainRids()

Get the domain map of all of the accessible domains with their corresponding RID information.

Returns:

- KeyValueSet (see KeyValueSet)

Raises:

- Error - if any error occurred. If the domain map construction is not complete, the code will be TRY_AGAIN.

networkChange()

Notify adclient that there was a network change on the system.

Returns:

- code as (int). If success, the code value will be CODE_SUCCESS (see Code constants)

ping()

Test the connection to the agent.

Returns:

- code as (int). If success, the code value will be CODE_SUCCESS (see Code constants)

getKerberosName(name, useSamName)

Get the Kerberos principal name of a user.

Parameters:

- name (str) - user name
- useSamName (int) - TRUE will use sAMAccount name (see Boolean constants)

Raises:

- Error - if any error occurred

authValidateAccount(name, flags)

Check a user account to see if any logon restrictions currently apply.

Parameters:

- name (str) - user name
- flags (int) - validate flags (see Validate Flag constants)

Returns:

- code as (int). If success, the code value will be CODE_SUCCESS (see Code constants)

authValidatePlainTextUserNonCDC(name, password)

Validate a non-DirectControl managed user.

Parameters:

- name (str) - user name
- password (str) - user password

Returns:

- code as (int). If success, the code value will be CODE_SUCCESS (see Code constants)

authValidatePlainTextUser(name, password)

Validate a user and password using Kerberos.

Parameters:

- name (str) - user name
- password (str) - user password

Returns:

- code as (int). If success, the code value will be CODE_SUCCESS (see Code constants)

systemHealthInfo(refresh=FALSE)

Return information about DirectControl's system health.

Parameters:

- refresh (int) - if FALSE, return information from last API call. If TRUE, send a probe to collect updated information. (See Boolean constants)

Returns:

- KeyValueSet (see KeyValueSet)

Raises:

- Error - if any error occurred

getForestList(flags)

Get the trusted forest information list.

Parameters:

- flags (int) - flags (see Get DC Flag constants)

Returns:

- ObjectList (see ObjectList)

Raises:

- Error - if any error occurred

getDomainList(flags)

Get the trusted domain information.

Parameters:

- flags (int) - flags (see Get DC Flag constants)

Returns:

- ObjectList (see ObjectList)

Raises:

- Error - if any error occurred

getDCInfo(name)

Get Information about a specific domain controller (DC).

Parameters:

- name (str) - name of the domain controller

Returns:

- Object (see Object)

Raises:

- Error - if any error occurred

getDomainControllers(name, flags)

Get a list of domain controllers for specific domain.

Parameters:

- name (str) - name of the domain
- flags (int) - flags (see Get DC Flag constants)

Returns:

- StringSet (see StringSet)

Raises:

- Error - if any error occurred

getAuditLevel(name)

Get audit level of a user.

Parameters:

- name (str) - user name

Returns:

- audit level as (int) (see Audit Level constants)

Raises:

- Error - if any error occurred

Throw Error exception in case of error.

Error Object Methods

The base class of Error is the Python Exception class.

message()

Returns a message as a string

Returns:

- message as (str) (see Audit Level constants)

code()

Returns code

Returns:

- code as (int) (see Code constants)

Pycapi Module Constants

This section lists out the different constant values that can be used with the Pycapi module.

Boolean Constants

TRUE	1

FALSE	0

Code Constants

CODE_SUCCESS	0
CODE_FAILURE	1
CODE_NOMEM	2
CODE_BAD_OPTION	3
CODE_BAD_PARAM	4
CODE_BAD_SESSION	5
CODE_LRPC_FAILED	6
CODE_NO_MORE	7
CODE_NO_SUCH_ATTR	8
CODE_NO_SUCH_OBJECT	9
CODE_SERVER_UNREACHABLE	10
CODE_SEARCH_IN_PROGRESS	11
CODE_BAD_VERSION	12
CODE_INVALID_USER	13
CODE_INVALID_PASSWORD	14
CODE_ACCOUNT_LOCKED	15
CODE_PASSWORD_EXPIRED	16
CODE_PASSWORD_POLICY_NOT_MATCHED	17
CODE_PASSWORD_CHANGE_REJECTED	18
CODE_ACCOUNT_EXPIRED	19
CODE_ACCOUNT_DISABLED	20
CODE_WORKSTATION_DENIED	21
CODE_PERMISSION	22
CODE_BAD_PACKET	23
CODE_BAD_DATA	24

CODE_NOT_JOINED	25
CODE_VALUE_NOT_SET	26
CODE_IO_ERROR	27
CODE_SYS_ERROR	28
CODE_NO_SYS_ERROR_INFO	29
CODE_WRONG_DATA_TYPE	30
CODE_MULTI_VALUE	31
CODE_NO_ADCLIENT	32
CODE_LOGON_FAILURE	33
CODE_NOT_GROUP_MEMBER	34
CODE_FOREIGN_DOMAIN	35
CODE_NOT_FOUND	36
CODE_EXISTS	37
CODE_TRUST_ERROR	38
CODE_ACCOUNT_LOGON_HOURS	39
CODE_ACCOUNT_WORKSTATION	40
TRY_AGAIN	41
CODE_NO_DNS	42
CODE_BAD_COMPUTER_OBJECT	43
CODE_ACCOUNT_RESTRICTION	44
CODE_ALREADY_JOINED	45
CODE_CLIENT_DISCONNECTED	46
CODE_GROUP_POLICY_NOT_FOUND	47
CODE_INVALID_CONTAINER	48
CODE_NAME_MATCHES_DC	49
CODE_NETWORK_ERROR	50
CODE_OUT_BOUND_TRUST	51
CODE_PROCESS_AUTHENTICATION	52

CODE_UNKNOWN	53
CODE_ZONE_ACCESS_PERMISSION	54
CODE_IN_ANOTHER_DOMAIN	55
CODE_FIPS_NONCOMPLIANT	56
CODE_BLOCKED	57
CODE_REENTERED	58
CODE_PASSWORD_DID_CHANGE	59

Error System Constants

ERR_SYS_NONE	0
ERR_SYS_KERBEROS	1
ERR_SYS_LDAP	2
ERR_SYS_NTSTATUS	3
ERR_SYS_BASE	4
ERR_SYS_AZMAN	5
ERR_SYS_DNS	6
ERR_SYS_NETWORK	7
ERR_SYS_GP	8
ERR_SYS_FIPS	9
ERR_SYS_EOL	10

Option Constants

OPT_UNIX_ONLY	0x00000001
OPT_CHECK_AD_FIRST	0x00000002
OPT_GROUP_MEMBERSHIP	0x00000004
OPT_UNIX_NAME	0x00000008
OPT_WINDOWS_NAME	0x00000010

OPT_APPLY_OVERRIDES	0x00000020
OPT_ZONE_SEARCH	0x00000040
OPT_AUTO_RECONNECT	0x00000080
OPT_AUTH_VALIDATE_ACCOUNT	0x00000100
OPT_CREATE_KRB5_CACHE	0x00000200
OPT_NO_CACHE	0x00000400
OPT_REFRESH_MEMBERSHIP	0x00000800
OPT_AUTH_VALIDATE_ACCT_PREFER_CACHE	0x00001000
OPT_LOCATE_ALL_SERVICES	0x00002000

Object Type Constants

OBJTYPE_USER	1
OBJTYPE_GROUP	2
OBJTYPE_COMPUTER	3

AD Category Constants

AD_CATEGORY_GROUP	"Group"
AD_CATEGORY_USER	"Person"
AD_CATEGORY_COMPUTER	"Computer"
AD_CATEGORY_CONTAINER	"Container"
AD_CATEGORY_ORGUNIT	"Organizational-Unit"
AD_CATEGORY_SCP	"Service-Connection-Point"
AD_CATEGORY_CLASS_STORE	"Class-Store"
AD_CATEGORY_FSP	"Foreign-Security-Principal"
AD_CATEGORY_ANY	""

Get DC Flag Constants

GETDC_FLAGS_GET_ALL	0x00000001
GETDC_FLAGS_WRITABLE	0x00000002
GETDC_FLAGS_NO_LIVE_TEST	0x00000004
GETDC_FLAGS_DONT_READ_CACHE	0x00000008
GETDC_FLAGS_IGNORE_KSET	0x00000010
GETDC_FLAGS_DEEP_SWEEP	0x000000020
GETDC_FLAGS_SPEED_SORT	0x000000040
GETDC_FLAGS_ANY_SITE	0x000000080

AD Attribute Constants

AD_ATTR_USERNAME	"name"
AD_ATTR_USER_PRINCIPAL_NAME	"_userPrincipalName"

Validate Flag Constants

VALIDATE_ACCT_LOCKOUT	0x00000001
VALIDATE_ACCT_DISABLED	0x00000002
VALIDATE_ACCT_EXPIRED	0x00000004
VALIDATE_PASSWD_EXPIRED	0x00000008
VALIDATE_WORKSTATIONS	0x00000010
VALIDATE_LOGON_HOURS	0x00000020
VALIDATE_ALL	0xffffffff

Audit Level Constants

AUDITLEVEL_NOTSET	-1
AUDITLEVEL_AUDITIFPOSSIBLE	0
AUDITLEVEL_NOAUDIT	1
AUDITLEVEL_AUDITREQUIRED	2

AUDITLEVEL_SYSRIGHTS	3

Pycapi Dictionary Objects

Some of the pycapi methods return objects, those are described below. A dictionary is a data type in Python that's used to store a set of key:value pairs.

Object	The Object is a dictionary object that stores the attributes of the object returned. For each item in the dictionary object, the key is a string, and the value is a list of bytes objects. If the attribute has only one value, the attribute will be a list with only one bytes object.
ObjectList	A list of objects.
StringSet	A list of strings.
KeyValueSet	A dictionary of strings.

About Delinea Management Services for Mac

With Delinea Management Services for Mac, you can use Active Directory to centrally manage authentication, policy enforcement, single sign-on (SSO), and user self-service for popular endpoint devices running Mac operating systems.

A key component of Delinea Management Services for Mac is the *DirectControl agent* for Mac computers. You must install the agent on each computer that you want to integrate with Active Directory and manage through Delinea Access Manager.

After you install the agent on a Mac computer, you can perform many administration and configuration tasks on the computer to enable the computer to work with Delinea Management Services and with Active Directory.

Intended Audience

This guide is intended for Mac system administrators.

Topics Covered in this Guide

The following topics are covered:

Installing the DC Agent Creating Home Directories Working with Macs Understanding Group Policies Setting Computer-based Policies Setting User-based Policies Configuring a Mac for Smart Card Login Troubleshooting Installing and Removing the Agent

The *Administrator's Guide for Mac* provides information about the administration and configuration tasks that you perform on a Mac computer after you install the agent so that you can manage users, groups, computers, and zones with Access Manager. Additional topics, such as installing the agent, optionally enrolling the computer in the Delinea Platform, and troubleshooting issues after the agent is installed are also covered.

Specific areas of focus are as follows:

- This guide provides installation instructions and step-by-step instructions for configuring Mac computers to join an Active Directory domain through Auto Zone, which creates one large zone for all Mac computers. Auto Zone requires minimal configuration and is appropriate for most Mac environments. If your environment is larger, or more complex, and doesn't easily fit into Auto Zone, you must consult the *Planning and Deployment Guide* for detailed information on how to move your Mac users and computers to Active Directory and use Delinea zones to structure your environment.
- This guide explains how to handle issues and tasks that are specific or unique to a Mac environment.

This guide does not cover planning or Access Manager tasks handled through the Access Manager console. For more information about those topics, see Where to go for more information.

This guide assumes you have a working knowledge of performing administrative tasks in a Mac environment.

This section explains how to install the DirectControl Agent for a Mac computer.

Preparing to Install the DirectControl Agent for Mac

You must install the DirectControl Agent for Mac on each computer that you want to manage through Delinea and Active Directory. You can check the *Release Notes* included with the software, or visit the Delinea Web site (scroll to **Supported Platforms** and click the **Details** tab) to verify that each computer where you plan to install is running a supported version of the mac operating system.

> **Note**: The installation package also contains a utility, ADCheck, which verifies that each of your Mac computers is ready for installation of the DirectControl agent. ADCheck confirms that a computer is running a supported OS, has sufficient disk space to install the DirectControl agent, and that the domain you intend to join has functioning domain controllers and DNS servers. Information about running ADCheck is included in this documentation.

Installing the Agent on Apple M1 Mac Computers

Depending on whether you using the graphical installer or the command line version to install the DirectControl Agent for Mac on Apple M1 Mac computers, you may need to install some additional software.

- If you install the DirectControl Agent for Mac on an Apple M1 Mac computer using the graphical user interface, you might be asked to install Rosetta.

  Click **Install**, then enter your user name and password to allow installation to proceed.

  For more information, see https://support.apple.com/en-us/HT211861.

- If you install the DirectControl Agent for Mac on an Apple M1 Mac computer using the install.sh script, or by installing remotely, you might need manually install Rosetta first. Please run the following command with root privileges to install Rosetta 2:

  ```
  /usr/sbin/softwareupdate --install-rosetta --agree-to-license
  ```

Verifying DirectControl Agent for Mac Installation Prerequisites

Before installing the DirectControl Agent for Mac on your Mac computers, be certain that you or another administrator has installed Delinea Management Services on a Windows computer in the domain. Delinea Management Services includes the Access Manager Console, which is the primary management console for performing ongoing operations, including the application of group policies. Always install this console unless you are installing and running Delinea Express for Linux and UNIX, which does not contain a console component.

For information about other Delinea Management Services components, such as Zone Provisioning Agent, see the *Planning and Deployment Guide* and the *Administrator's Guide for Linux and UNIX*.

Deciding When and How to Join a Domain

Following installation, you will be prompted to join a domain. Whether to join a domain depends primarily on how you intend to join. Delinea provides two ways to join a domain:

- Through Auto Zone, which is the recommended method for installations with 1500 or fewer users. When joined through Auto Zone, all users and groups defined in Active Directory for the forest — as well as all Active Directory users defined in a forest with a two-way, cross-forest trust relationship to the forest of the joined domain — automatically become valid users and groups on the Mac computer.
- By connecting to a specific Delinea zone, which is the recommended method for installations with 1500 or more users, or for installations in which fine-tuned access control is needed. A zone is similar to an Active Directory organizational unit (OU) and allows you to organize the computers in your organization in meaningful ways to simplify account and access management and the migration of information from existing sources to Active Directory.

The assumption of this guide is that you are joining Auto Zone. After installation, you can follow the instructions to join the domain and with a few configuration steps all your Active Directory users will be able to log into this computer.

> **Note**: If you have a set of Apple Open Directory users, you should migrate them following installation but before joining a domain.

On the other hand, if your environment requires a zone structure, you must create that structure before joining a domain. Therefore, after installing the DirectControl agent, consult the *Planning and Deployment Guide* and the *Administrator's Guide for Linux and UNIX*, which explain in detail how to plan, create, and maintain an Active Directory installation of non-Windows computers with Server Suite.

Installing the DirectControl Agent

The DirectControl Agent for Mac can be installed in several different ways. The procedure in this section shows how do so by double-clicking the Delinea Installer package (DMG) and following the instructions displayed on the screen. This installation method is recommended for most users when installing on a single computer or a limited number of computers.

When you use the Delinea package installer, you will be prompted to join the domain. You may also join the domain after installation using either the adjoin command-line program or the Delinea Directory Access plug-in.

Delinea provides a number of other ways to install the DirectControl Agent for Mac

- By executing the DirectControl Agent for Mac installation script, install.sh in a Terminal window on a Mac computer and following the instructions displayed by the script.

  If you are an experienced UNIX administrator and are familiar with UNIX command-line installations, running install.sh is a good method to use. When you install using the install.sh script, you can automatically join an Active Directory domain as part of the installation process; see Installing Using the install.sh Script for details.

- By installing remotely, without user interaction, using Apple Remote Desktop. This is a good method to use if you are using Apple Remote Desktop for software distribution. With Apple Remote Desktop you can add pre- and post-installation scripts that allow you to join the remote computer to a domain after installation; see Installing Silently on a Remote Computer for details.

To install the DirectControl Agent for Mac on a Mac computer using the graphical user interface:

1. Before installing the DirectControl Agent for Mac, disable Apple's built-in Active Directory plug-in, and remove Active Directory from the Authentication, and Contacts search paths. For more information, see Disabling the Apple Built-in Active Directory Plug-in.
2. In addition, be certain that the Apple Directory Utility is closed.
3. Log on with the Administrator account.
4. Navigate to the directory on the CD or your local network where the agent package is located. For example, if you are installing from the Delinea CD, open the MacOS directory.
5. Double-click the DMG file, for example:

   centrifydc-release-mac10.10-x86_64.dmg

6. Double-click ADCheck to open the ADCheck utility.

   ADCheck performs a set of operating system, network, and Active Directory checks to verify that the Mac computer meets the system requirements necessary to install the DirectControl Agent for Mac and join an Active Directory domain.

7. Enter the domain you intend to join with the Mac computer and click **AD Check**; for example:

8. Review the results of the checks performed. If the target computer, DNS environment, and Active Directory configuration pass all checks with no warnings or errors, you should be able to perform a successful installation and join the specified domain. If you receive errors or warnings, correct them

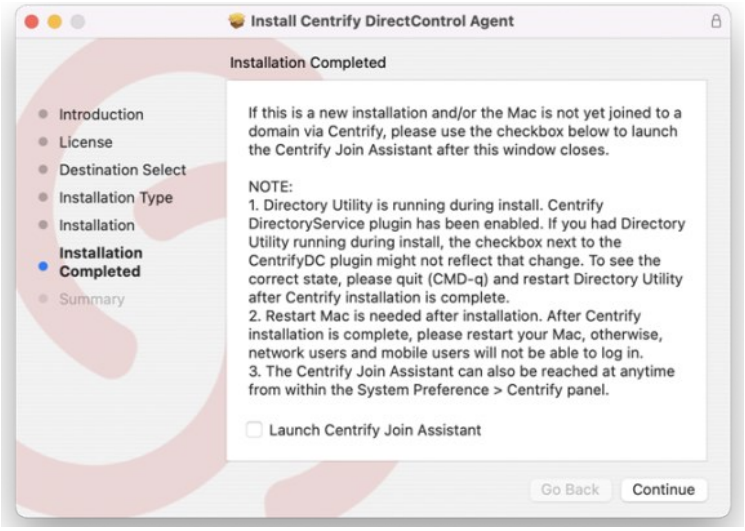
before proceeding with the installation; see the *Administrator's Guide for Linux and UNIX* for more information about ADCheck.

9. Double-click the DelineaDC package to open the Installer:

10. Review the information in the Welcome page, then click **Continue**.
11. Review or print the terms of the license agreement, then click **Continue**; click **Agree** to agree to the terms of the license agreement. Then click **Install** (note that you cannot change the volume on which the agent is installed — it must be on the same volume as Mac OS X).
12. If prompted, enter the administrator name and password, and click **Install Software** to install the DirectControl Agent for Mac.

If you see the following warning box, click **OK**. If you did not have Directory Utility running during the installation, you can ignore the warning. If Directory Utility was open, you can quit and restart it to show the correct status of the Delinea plug-in.

The installation process runs and presents the Installation Completed page once the DirectControl Agent for Mac is installed.



13. Select **Launch Delinea Join Assistant** if you want to join a domain, then click **Continue**.

    Note: If you know that you want to use Delinea zones in your environment, exit the installer now. You must create zones first, before you can join to one. Refer to Deciding When and How to Join a Domain for more information.

If you chose not to launch the Delinea Join Assistant before clicking Continue, the installer presents a summary indicating that the installation was successful. You can now close the installer.

If you chose to launch the Delinea Join Assistant, you can start the process of **Joining an Active Directory Domain** described in the next section.

Note: If the Mac system is MacOS 11 or later, you must configure full disk access for the DirectControl Agent for Mac before you join the system to an Active Directory domain.

Joining an Active Directory Domain

This topic shows how to use the Delinea Join Assistant to join a domain. To join a domain, you must be a domain admin or a domain user with permission to create computer objects. If necessary, your domain administrator can use the Delegation Wizard to delegate permission to create computer objects. Refer to Who Can Add a Workstation to a Domain for more information.

Note: Alternately, you may run the adjoin command-line utility, interactively or in a script, for each Macintosh computer you want to add to a

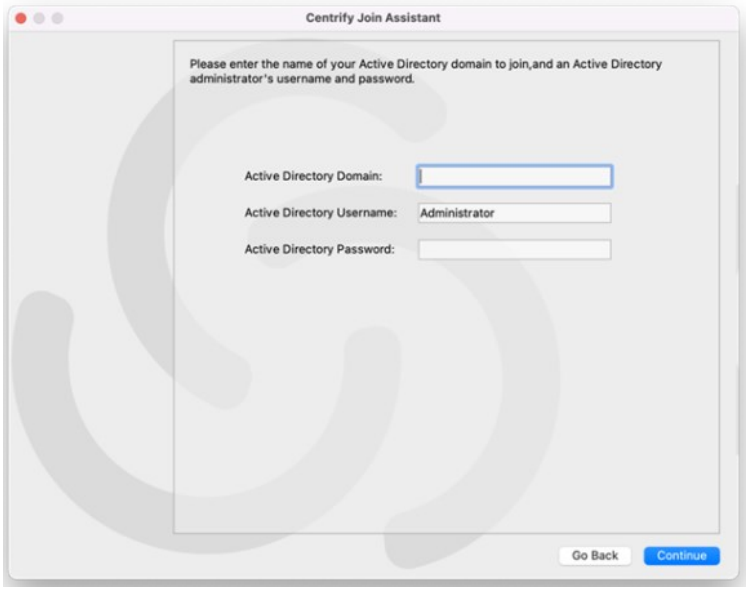
domain in the forest. See the *Administrator's Guide for Linux and UNIX* for details.

To join the Mac to a domain:

1. Launch the Delinea Join Assistant.

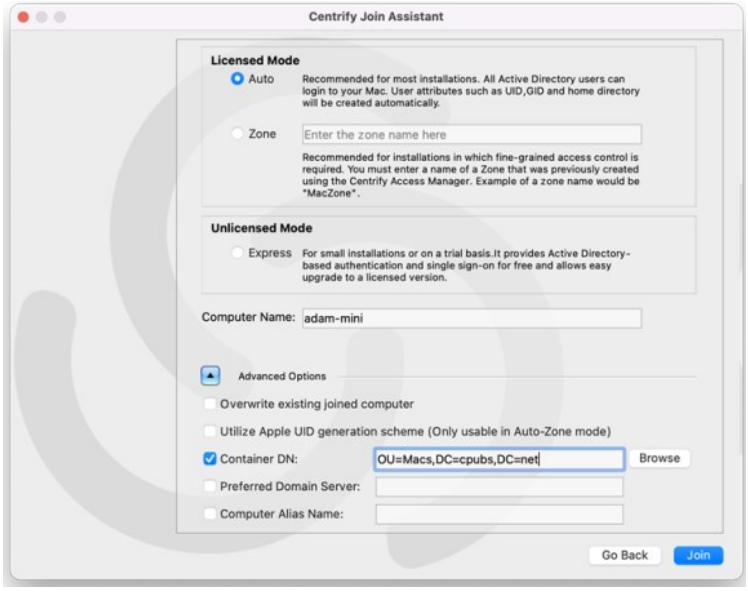
   There are two ways to launch the Delinea Join Assistant:

   - from the DirectControl agent installer, as described in Installing the DirectControl Agent for Mac.
   - click **Applications › Utilities › Delinea**, double-click **Delinea Join Assistant** to open it, then click **Continue** on the Welcome page



2. Enter the active directory domain that you want to join as well as administrator credentials for that domain, then click **Continue**.

   A page appears that allows you to select how to join the domain with an option to enroll in the Privileged Access Service.



3. Select from the following options:

Auto	Joins the computer through Auto Zone, which allows joining a computer with little or no configuration. This option is recommended for most installations.
Zone	Joins to the zone that you type in the box. Note that you must have created at least one zone before you can use this option.
Computer name	Defaults to the name of the computer on which you are running the join assistant, but you can change it if you want to use a different name for the local host in Active Directory.

Note: Enrollment is no longer supported.

4. (Optional) Click the arrow to expand the Advanced Options and select any Advanced Options that you want to use to join the device.

Overwrite existing joined computer	Overwrite the information stored in Active Directory for an existing computer account. This option allows you to replace the information for a computer previously joined to the domain. If there is already a computer account with the same name stored in Active Directory, you must use this option if you want to replace the stored information. You should only use this option when you know it is safe to force information from the local computer to overwrite existing information. Checking this option is the same as running the adjoin command with the `--force` option.
Container DN	Specify the distinguished name (DN) of the container or Organizational Unit in which you want to place this computer account. By default, computer accounts are created in the domain's default Computers container. Click **Browse** to browse Active Directory and select the container to use, or click Container DN and enter the name of the container in distinguished name format; for example, if the domain suffix is acme.com and you want to place this computer in the paris.regional.sales.acme.com organizational unit, you would type: `ou=paris, ou=regional, ou=sales` Checking this option is the same as running the adjoin command with the `--container` option.
Preferred Domain Server	Specify the name of the domain controller to which you prefer to connect. You can use this option to override the automatic selection of a domain controller based on the Active Directory site information. Checking this option is the same as running the adjoin command with the `--server` option.
Computer Alias Name	Specify an alias name you want to use for this computer in Active Directory. This option creates a Kerberos service principal name for the alias and the computer may be referred to by this alias. Checking this option is the same as running the adjoin command with the `--alias` option.

5. Click **Join**.

   Delinea Join Assistant informs you that you have successfully joined your Mac to your Active Directory domain at <mydomain.com>.

6. Click **Done** to close the Delinea Join Assistant.

   Your Active Directory users can now log on to the joined Mac computer, as described in Logging onto the Mac after Joining a Domain.

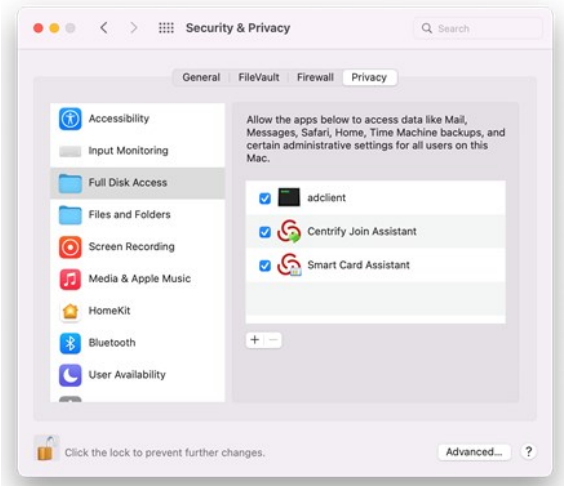
Configuring Full Disk Access for the DirectControl Agent for Mac

Due to a limitation of MacOS 11.x and MacOS 12.x, "Full Disk Access" is required for the DirectControl Agent for Mac. You can configure this yourself if you're an administrator on the computer, or you can set it by way of your MDM (Mobile Device Management) provider.

To configure full disk access as an administrator:

1. Log in to the Mac computer as an administrator user.
2. Open **System Preferences**.
3. Click **Security & Privacy.**

4. Click **Privacy**.
5. Click **Lock** and then enter the password or use TouchID to unlock.
6. In the left pane, scroll down and select **Full Disk Access**.
7. Click **+** (the plus button).
8. Press and hold these three keys together: Sshift + Command + G.
9. Enter the path "/usr/local/sbin/adclient" and click **GO**, then click **Open** to add the path.
10. Repeat step 7 and 8, then input the path "/Applications/Utilities/Centrify/Centrify Join Assistant.app" and click **GO**, then click **Open** to add the path.
11. Repeat step 7 and 8, then input the path "/Applications/Utilities/Centrify/Smart Card Assistant.app" and click **GO**, then click **Open** to add the path.
12. Click **Lock** again to lock the system preferences.



Configuring Full Disk Access Through Your MDM Provider

Contact your MDM provider for more information. Your MDM provider will need the following information:

```
% codesign -dv /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
Identifier=adclient
...
% codesign -dr - /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
designated => identifier adclient and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
certificate leaf[subject.OU] = "64CT837G5Z"
% codesign -dv /Applications/Utilities/Centrify/Centrify\ Join\ Assistant.app
Executable=/Applications/Utilities/Centrify/Centrify Join Assistant.app/Contents/MacOS/Centrify Join Assistant
Identifier=com.centrify.cdc.centrifyjoinassistant
...
% codesign -dr - /Applications/Utilities/Centrify/Centrify\ Join\ Assistant.app
Executable=/Applications/Utilities/Centrify/Centrify Join Assistant.app/Contents/MacOS/Centrify Join Assistant
designated => identifier "com.centrify.cdc.centrifyjoinassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"
% codesign -dv /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
Identifier=com.centrify.cdc.smartcardassistant
...
% codesign -dr - /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
designated => identifier "com.centrify.cdc.smartcardassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"
```

Configuring Full Disk Access for Apple Remote Desktop

If your organization uses Apple Remote Desktop to run any DirectControl Agent for Mac commands (such as adjoin, adleave, and so forth), you need to also set Full Disk Access for Apple Remote Desktop. You can do this either as an administrator user or through your MDM service, following the same procedures

as mentioned earlier.

If you're configuring full disk access as an administrator, the application path to add is as follows:

```
/System/Library/CoreServices/RemoteManagement/ARDAgent.app
```

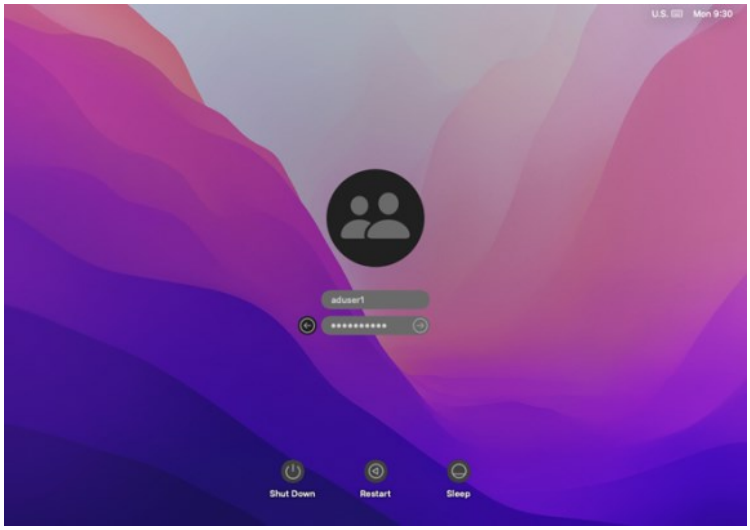
If you're configuring full disk access through your MDM provider, here's the information that your provider needs:

```
% codesign -dv /System/Library/CoreServices/RemoteManagement/ARDAgent.app
Executable=/System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/MacOS/ARDAgent
Identifier=com.apple.RemoteDesktopAgent
...
% codesign -dr - /System/Library/CoreServices/RemoteManagement/ARDAgent.app
Executable=/System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/MacOS/ARDAgent
designated => identifier "com.apple.RemoteDesktopAgent" and anchor apple
```

Logging onto the Mac After Joining a Domain

When using Auto Zone, all Active Directory users in the domain become valid users on a joined computer. To verify that the software is working properly, you can simply log into the Mac computer by using an Active Directory account.

On the Mac login screen, select **Other** and enter an Active Directory user name and password:



Upgrading The DirectControl Agent for Mac

In most cases, you can update agents on Mac computers by simply installing the new agent either directly or remotely on top of an existing agent. As a best practice, you should perform in-place upgrades using a local Mac administrative (admin) account or any other user account that has local administrative rights and reboot the computer after completing the upgrade. In most cases, you should not perform the upgrade while you are logged on as an Active Directory user in a currently active session.

In rare cases, you might be advised to run adflush to clear the Active Directory cache before performing an in-place upgrade. For example, if you are updating agents from version 4.x, or earlier, to 5.1.x, run adflush first to ensure a smooth upgrade. It is highly unusual for an upgrade to require you to leave and rejoin a managed Mac computer to the domain.

This section explains how to create different types of home directories for a Mac computer.

Understanding Home Directories

Whenever an Active Directory user logs in to a Mac computer, a home directory is created for the user. Mac provides three styles of home directory, which can be configured by an administrator to fit the type of user who will be using the computer, the type of computer, and the use to which the computer will be put. Auto Zone supports each of these styles:

- Local home directory — The user's home directory is created on the local computer in the Users folder with the user's login name (/Users/username).
- Network shared directory — The user's home directory is created on a network share.
- Portable home directory — The user's home directory is created on a network share and copied and synchronized to the local computer. This type of directory is also called a *mobile* home directory.

When you join a computer to a domain by connecting to Auto Zone, the home directory is created based on the following:

- Active Directory user settings; for example, an administrator can specify a network home directory in the Profile for an Active Directory user.
- Auto Zone default values; by default, Auto Zone is configured to support the creation of home directories in the Users folder on the local computer.
- Auto Zone parameters set in the configuration file, /etc/centrifydc/centrifydc.conf by an administrator or by a group policy. See the *Configuration and Tuning Reference Guide* for a description of all Auto Zone parameters.

The following sections explain in detail how to set up each type of user home directory.

Configuring a Local Home Directory

In general, you do not need to explicitly configure local home directories for your Active Directory users because Auto Zone is configured to work for Active Directory users exactly as if they were local users. That is, by default, an Active Directory user who logs in to a Mac computer that is joined to a domain through Auto Zone is given a local home directory at /Users/username. For example, for a user, Glen Morris, whose login name is gmorris, the local home directory is set to: /Users/gmorris.

Although it isn't necessary to explicitly configure the agent for local home directories, in some situations you might want to do so. For example, if a Windows user has a local home directories defined in their Active Directory profile, that home directory will be assigned when the user attempts to log in and may prevent the user from logging in. The agent provides a configuration parameter (auto.schema.use.adhomedir)that you can set to ignore home directories in an Active Directory profile and always set the home directory to the default (/Users/username).

To explicitly configure a computer for local home directories:

1. On the Mac computer, edit the configuration file, /etc/centrifydc/centrifydc.conf.
2. Add the following two parameters:

   auto.schema.use.adhomedir: false

   auto.schema.homedir: /Users/%{user}

   - Setting auto.schema.use.adhomedir to false configures the local computer to ignore any home directories that are set for users in Active Directory. This parameter is set to true by default.
   - Setting auto.schema.homedir: /Users/%{user} configures the local computer to set the home directory to /Users/username, where *username* is the user logon name defined in the user's Active Directory account. Note that this parameter is set to this value by default on all Mac computers.

     Note: If you plan to configure network-home or portable-home directories for this computer, you must set auto.schema.use.adhomedir to true, the default value, otherwise, the agent will ignore the network home directories that you specify for users in Active Directory.
3. Save and close the file.

Configuring a Network Home Directory

For each user whom you want to have a network home directory, you must specify the location in Active Directory.

Note: In earlier releases you had to first create a network home directory for a user if you planned to also create a portable home (mobile home) directory for that user. With the current release, you can create portable home directories for users without first creating network home

directories for those users.

To configuring a network home directory for a user connected to Auto Zone:

1. Create a network share to host the home directory.

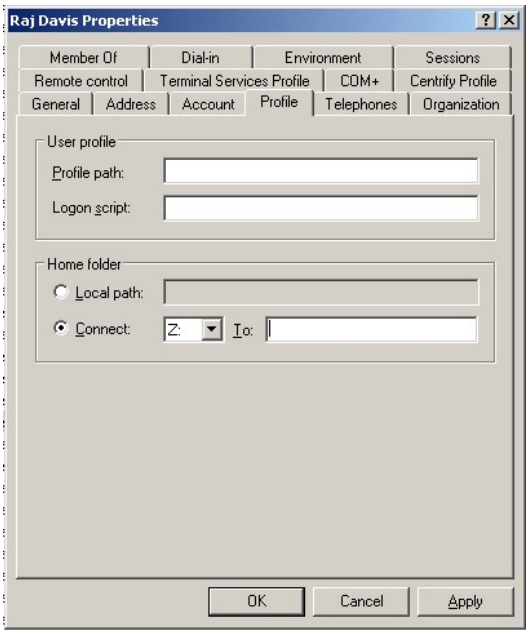
   For example, on the dc-demo server (acme.com domain), create a network share called MacUsers.

   You must assign appropriate permissions to the network shared directory so the Active Directory account is able to write to the user's home directory. One way to do this is to assign read/write permissions to Authenticated Users on the network share. Each home directory that is created inherits permission from the network share so the account of the logged-in user is granted write permission its network home directory. See Setting Shared Directory Permissions for more details about properly setting and fine-tuning network share permissions.

2. On a domain controller in the forest to which the Mac OS computer is joined, open Active Directory Users and Computers.

3. Select **Users**, select the user, then right-click the user and click **Properties**.

4. Click the **Profile** tab, then under **Home folder** select **Connect**.



5. In **Connect...To** type the location of the share you created in Step 1 by using the following format:

   ```
   //*Server/share/path
   ```

   For example:

   ```
   //dc-demo.acme.com/MacUsers/rdavis
   ```

6. Click **OK** to save the user profile.

7. (Optionally) By default, the agent is configured to use the Active Directory home folder if one is specified in a user's profile. However, to be explicit, you can edit the configuration file and add the following parameter:

   ```
   auto.schema.use.adhomedir: true
   ```

   Save and close the file.

8. Specify the type of share to mount for the network home directory on the Mac computer, SMB, or AFP.

   By default, the Mac computer will attempt to mount an SMB share for the network home. If you specified an AFP share, you must set the following parameter in the configuration file:

auto.schema.remote.file.service:AFP

Or enable the **Computer Configuration > Policies > Centrify Settings > DirectControl Settings > Adclient Settings > Auto Zone remote file service** group policy to specify SMB (the default) or AFP for all Mac computers.

9. Optionally, if you want the network home directory to be mounted automatically on the user's computer, enable the following group policy: **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Automount Settings > Automount user's Windows home**.

When the specified user next logs onto the Mac computer, the home directory will be created on the specified share. On the Mac computer, you should see the server and share under **SHARED** in the Finder.

Configuring a Portable Home Directory

You can create a portable home directory for a user and synchronize that directory with the share defined in the user's Centrify Profile. You can synchronize to /SMB/, /AFP/, or /Network/Servers (NFS) shares.

Advantages of a Portable Home Directory

- If a user does not have a portable home directory and the computer becomes disconnected from the domain controller (and therefore disconnected from Active Directory), the user can log in with Active Directory credentials only if the user's information exists in the Centrify cache. If there is any issue with the Centrify cache (for example, if the adflush --force command was issued to flush the cache immediately before the computer was disconnected from the domain), Active Directory users cannot log in unless they have portable home directories.
- Active Directory users without portable home directories are required to log in at least once in connected mode to populate their account information in the Centrify cache. If the computer is not connected to the domain controller, the Centrify cache is not updated with the initial set of Active Directory user data, and Active Directory users cannot log in. You use group policies to configure synchronization. These group policies perform the same function as the Mobility preferences that you can manage through Workgroup Manager.

The following sections describe the process of specifying the options for creating mobile accounts, and for specifying the options for synchronizing mobile accounts with the network home directory.

Before you begin you should have the following in place:

- A Group Policy Object that applies to a domain or OU that includes Mac users.
- A good understanding of the synchronization rules that you want to apply. The procedures in the following sections explain the group policies and options that you can enable, but you should consult the Mac OS X Server documentation for strategies about which options to apply.

This section describes the unique characteristics or known limitations that are specific to using Delinea Management Services on a Mac computer.

Specifying the Macintosh User's Home Directory Location

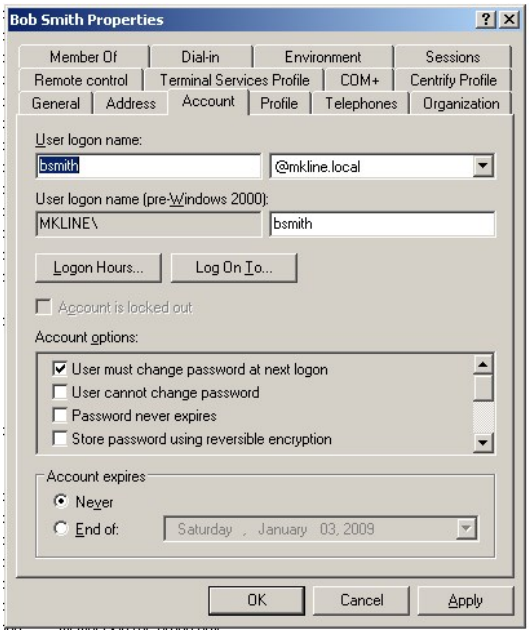
If you configure NFS, SMB, or AFP network file sharing for your Mac OS X computers, you can automatically mount and log on to file shares using Active Directory credentials.

To enable Mac OS X users to log on to file shares when the network is configured with NFS, SMB, or AFP network sharing:

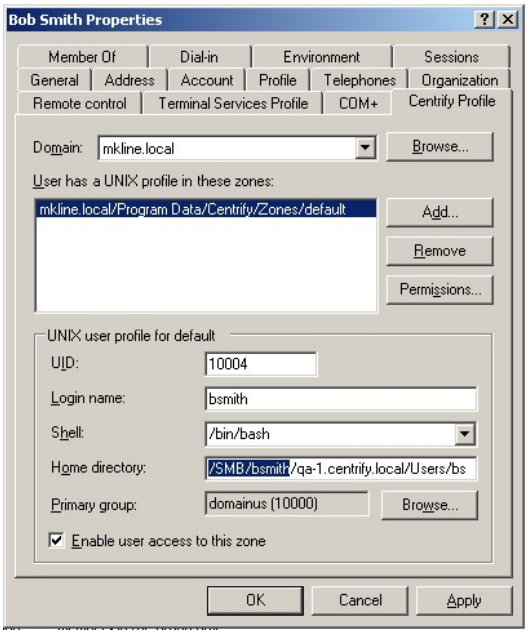
1. Open Active Directory Users and Computers or the Access Manager console.
2. Select the user account for which you want to enable automounting, right-click, then click **Properties**.
3. Click the **Delinea Profile** tab and set the **Home directory** path to use one of the following formats:
    - /Users/user_login_name to set the user's home directory to the default home directory location for all user home directories on Mac OS X computers.
    - /SMB/server_name/share[/path] to automount a file share on the SMB *server_name* you specify. Be certain to use the fully-qualified domain name for *server_name*, or the IP address. The short name does not work. For example:
      /SMB/myHost.acme.com/Users/isuzuki
    - /SMB/unix_username/server_name/share[/path] to automount a file share when you are using Fast User Switching on the SMB *server_name* you specify. Be certain to use the fully-qualified domain name for *server_name*, or the IP address. The short name does not work. For example:
      /SMB/isuzuki/myHost.acme.com/Users/isuzuki
    - /AFP/server_name/share[/path] to automount a file share on the Apple *server_name* you specify.
    - /AFP/unix_username/server_name/share[/path] to automount a file share when you are using Fast User Switching on the Apple *server_name* you specify.

      In specifying the remote SMB or AFP file share, you must use the uppercase letters SMB or AFP at the beginning of the path. If you use lowercase letters (smb or afp), automounting fails.

      Note: If you plan to use Fast User Switching to switch between Active Directory users on the same computer, you should use the /SMB/unix_username/server_name/share[/path] or /AFP/unix_username/server_name/share[/path] format to specify the user's home directory to prevent conflicts between users logging on using the same share. If you want to automount a share on an Apple file server using the Apple File Protocol (AFP), however, you must use Delinea 3.0.1 or later.
4. In Step 3, if you specified a network directory, make sure the Active Directory user logon name (pre-Windows 2000), also known as the samAccountName, matches the Mac login name (UNIX name). Otherwise, the login is not guaranteed to work on all Mac systems. The name must be eight characters or fewer because the UNIX name is automatically truncated to eight characters and won't match if the Active Directory name is longer. The Active Directory name is defined on the **Accounts** tab. To see an example, open the **Properties** page for a user and select **Account**.



Then select the **Delinea Profile** tab to see the UNIX name.



5. For the shared directory you specified in Step 3 (for example, Users), set 'full' permissions for authenticated users. See the section, Setting Shared Directory Permissions, for details on how to do this.

6. Verify that the computer on which the shared directory resides is configured on the DNS server with forward and reverse lookup zones by running the following commands in a terminal window:

   ```
   nslookup computerName.domainName
   ```

   for example:

   ```
   nslookup QA1.acme.com
   Server: acme.com
   ```

Address: 192.168.1.139

Name: QA1.acme.com

Address: 192.168.1.139

nslookup ipAddress

for example:

nslookup 192.168.1.139

Server: acme.com

Address: 192.168.1.139

Name: QA1.acme.com

Address: 192.168.1.139

If you get an error message such as this:

Can't find server name for address 192.168.1.139

it means a reverse lookup zone is not configured for the specified server. To configure DNS forward and reverse lookup zones, see the Microsoft Support Article 816518.

Populating the Home Directory on a Network Share

If you configure users to automount a network share when they log on, you must determine whether a home directory already exists on the network share for those users. If the individual user's home directory does not exist on the network share, Access Manager creates the home directory automatically the first time the user logs on.

Note: For NFS shares, Access Manager cannot create the home directory on the network share, so you must create the directory before users log in for the first time.

For example, assume you have defined the home directory in a user's Delinea Profile as: /SMB/demo-dc.acme.com/home/thomas, indicating that there is an SMB share on the server demo-dc and a shared folder named home where the user thomas has permission to list and create folders.

Note: For the server name, be certain to use the fully-qualified domain name, such as demo-dc.acme.com, and not the short version demo-dc.

When the zone user thomas logs on for the first time, Access Manager creates the new home directory thomas and populates it with the standard Mac OS X files and folders.

If the home directory specified in the Delinea Profile for a zone user exists prior to the user's first logon, Access Manager assumes that the directory is valid and contains the appropriate files, and it does not populate the directory with additional Mac-specific folders.

Defining a Home Directory in the Active Directory Profile

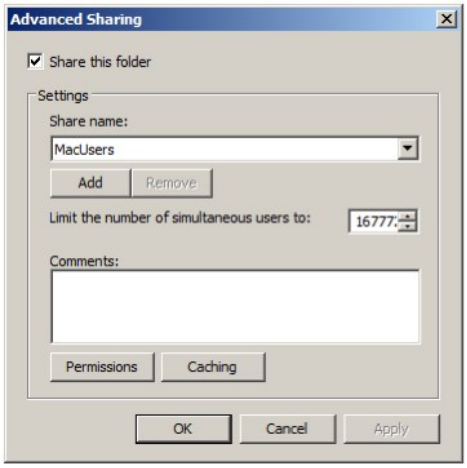
When you are configuring a network home directory for remote Mac users, the home directory is created automatically when users first log; it should not exist prior to that initial log on unless you want to prevent Access Manager from creating the home directory. Therefore, you should not define a home directory connection point in the Profile properties for new Active Directory users or new mobile user accounts. Instead, you should allow Access Manager to create and populate the remote home directory. However if you need to synchronize a network home directory from a local home directory as part of your migration process, the network home directory must exist prior to migration. If you are synchronizing from a local home directory to a remote share, you can create the remote home directory manually, or click the **Profile** tab and set the connection path.

Setting Shared Directory Permissions

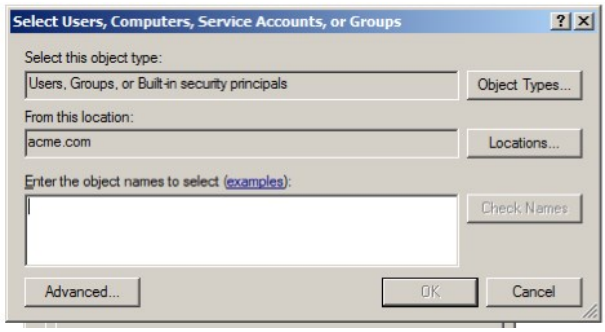
All users who are set up with a network home or portable home directory must have proper permissions to the shared directory in which the home directories are created. Initially, you can provide access to the shared directory through the Windows built-in security group, Authenticated Users. Later, you can fine tune permissions for this group based on your company's file-sharing needs. For example, if an administrator pre-creates home directories for each user before they log in, users only need Read access to the shared directory to access their home directories.

To set permissions for the shared directory for network home and portable home directories:

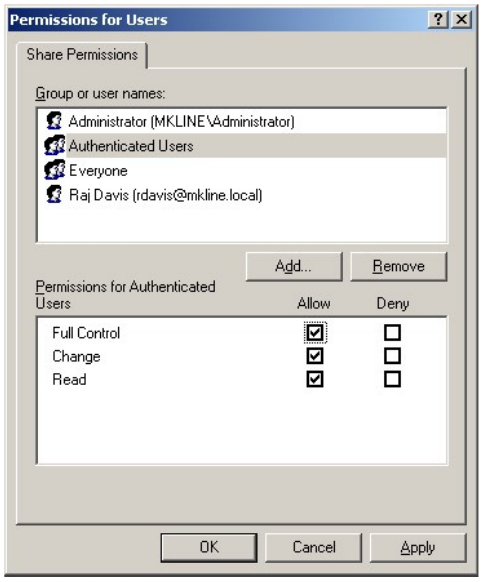
1. On the network share computer, select the directory to share (for example, MacUsers). Right-click, click **Properties** and click the **Sharing** tab; then click **Advanced Sharing**; for example:



2. Make sure **Share this folder** is selected. Click **Permissions**, then click **Add**:



3. Type auth and click **OK** to return the Authenticated Users group. Select **Authenticated Users**, then click **Allow** for **Full Control**. Click **OK** to set permissions for authenticated users, then click **OK** again to close the properties page.



4. Verify that Authenticated Users have proper permissions on the **Security** tab as well as on Share Permissions. Ordinarily, these permissions are applied automatically because the Active Directory Users group, which includes authenticated users, inherits Full Control to the shared folder, but if permissions were altered on the Security tab and they are insufficient, users may be unable to log in.

Click the **Security** tab and select **Authenticated Users** (or if it is not already in the Group or user names box, click **Add** to add it).

5. Select **Full control** and click **OK** to save and close the Properties page.

   Assigning permissions to Authenticated Users on the network home share directory means that each home folder will inherit permissions that enable logged-in users to access their home directories. It also means that every user will have access to every other user's home directory. To change this access, you can set permissions on the individual home directories. For information about fining tuning permissions for individual users, see the next section, **Limiting Users Access to Other Users' Home Folders**.

Limiting Users Access to Other Users' Home Folders

The previous section explained how to assign permissions to a network-home shared folder, which are consequently inherited by the home folders created in the shared folder. Because permissions are inherited, each user has equal access to every other user's home folder. This section explains how to fine tune permissions to limit user's access to their own home folder.

To limit users access to their own home folder:

1. Select the network share you assigned permissions to in the previous section.
2. Select one of the user home directories in the network share.
3. Click the **Security** tab.
4. Click **Advanced** and **Change Permissions**.
5. Deselect **Include inheritable permissions from the object's parent**.
6. Click **Remove** when prompted.
7. Click **Add**.
8. Type users and click **Return**.
9. Select the following permissions for Users:
   - Traverse folder / execute file
   - Read Attributes
   - Read Extended Attributes
   - Create files / Write Data
   - Create Folder / Append Data
10. Click **OK**, and **OK** again until you have saved all open dialogs and closed the Properties page.

Enabling Users to Manage Their Print Queues

On Mac computers, Delinea Active Directory users are unable to manage their own print jobs. For example, if they attempt to pause, stop, or resume one of their own print jobs, they are prompted to supply the name and password of a user in the "Print Operator" group, otherwise, they cannot continue. Delinea supplies the group policy, *Map zone groups to local group*, that you can use to enable all Mac users authenticated through Active Directory to manage their printers.

This policy gives members of a specified zone group (an AD group, or AD group that has been added to a Delinea zone) the privileges that belong to members of a local group on the local group. For example, as explained in the following procedure, mapping an AD group to the local _lpoperator and _lpadmin groups, provides members of the AD group with the privileges to manage print jobs on the local Mac computer when they log in.

To map a zone group to local _lpoperator and _lpadmin groups:

For purposes of illustration, this procedure asks you to create a **MacPrint** group and then add specific users to the group to provide them with printing privileges on Mac computers. You could also map an existing AD group to the local _lpoperator and _lpadmin groups, or create a new group with a different name.

1. On a Windows computer, open Active Directory Users and Computers
2. Select **Users**, right-click and select **New > Group**.
3. Enter a name for the group, such as MacPrint and select **Global** and **Security**.
4. Double-click the group and select the **Members** tab
5. Click **Add** and select the AD users who you want to provide with printing privileges on Mac computers.
6. Open the Access Manager Console
7. Expand the zone hierarchy as well as the zone containing Mac computers.
8. Expand **UNIX Data**, select **Groups**, then right-click and select **Create UNIX Group**.
9. Find and select the AD group you created (MacPrint) and click **OK** to add it to the zone.
10. Open the Group Policy Management Editor and select the GPO that you use for Mac OS X computers.
11. Click **Computer Configuration > Policies > User Configuration > Policies > Delinea Settings > Mac OS X Settings > Accounts**.

12. Double-click **Map zone groups to local group**.
13. Click the **Policy** tab and click **Enabled**.
14. Click **Add** and do the following:
    1. In **Local Group**, type _lpoperator to add the printer operators group.
    2. In **Zone Group** click **Browse**.
    3. Find and select the AD zone group you created (MacPrint), then click **OK** to map MacPrint to the printer operators group.
    4. Click **Add** again and in **Local Group** type _lpadmin to add the printer admin group.
    5. In **Zone Group**, click **Browse** then find and select MacPrint again to map MacPrint to the printer admin group.
15. Click **OK** to save the policy.

The first time users attempts to manage their printer, for example by pausing the printer, they will be prompted for credentials for a user in the "Printer Operator" group. They can simply enter their own name and password. Subsequently, they can manage the printer without supplying credentials.

Setting Up Authenticated Printing

In a Windows Active Directory environment that requires authentication for printing services, Mac users who are already authenticated must provide credentials again when using a Windows network printer. To provide single-sign on when using printers, the Delinea DirectControl Agent for Mac includes an authenticated printer plug-in that enables users to send print jobs to printers on the Windows network without requiring them to enter credentials again. This plug-in uses the user identifier (UID) of the user printing a job to find the user account to authenticate, then validates the user's Kerberos credentials through Active Directory. If the user's credentials are not available, the print job will fail.

Understanding Printing on Mac OS X

Mac uses the Common UNIX Printing System (CUPS) to manage printing services. Although you can access the CUPS facility directly to manage printers, in general you do not need to do so. Printers are managed through the Print and Scan system preference, which uses the CUPS facility. For example, when you add a printer through Print and Scan, the CUPS facility does the following:

- Creates a Postscript Printer Description (PPD) file that defines the printer. The file is given the name of the printer and resides in the /etc/cups/ppd directory; for example, /etc/cups/ppd/laserjet2.ppd.
- Modifies the CUPS configuration file, /etc/cups/printers.conf, with information about the new printer.

One method to set up authenticated printing for all Mac computers in your environment is to configure an authenticated printer on one (template) computer, then export the files that CUPS creates to define this printer (*printerName*.ppd and printers.conf) to each of your Mac computers. You can use group policy to export these files to all your Mac computers.

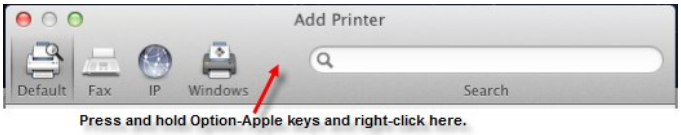
You can also configure printing directly with CUPS commands.

To set up authenticated printing for multiple printers using the Delinea plug-in, first identify the printer to configure, including the server that hosts it; for example, HPLaserJet2.@dc01.

1. On the Mac computer that you will use to define an authenticated printer template, open **System Preferences > Print & Scan** (**Print & Fax** on older systems), then click the plus sign (+) and select **Add Other Printer or Scanner**.

2. Double-click the **Advanced** icon in the toolbar.

   Note: If the Advanced option is not showing, press and hold the **Option** and **Apple** keys and right-click in the open area in the toolbar next to the Windows icon and select **Customize Toolbar**. Drag the Advanced icon to the toolbar and click **Done**. Then double-click it.



3. Scroll in the **Type** drop-down list and select **Windows Printer via Delinea** from the list.

   Note that after you make this selection, the URI scheme in the Device URI window changes to cdcsmb://, which specifies the Delinea plugin.

4. Type the complete URI specification for the printer in the form:

   scheme://servername/sharename

   for example:

```
cdcsmb://printserver.acme.com/hplaserjet2
```

Note: A URI specification does not accept spaces. If the printer share name contains spaces, you must replace them with %20 (ASCII code for space); for example, to specify the **HP Color LaserJet 4** printer:

```
cdcsmb://printserver.acme.com/HP%20Color%20LaserJet%204
```

5. Type a name for the printer; for example HPLaserJetMac.

   When you type the URI for the printer, the first part of the name automatically appears in the **Name** field. You can change that name now. This is the name that will appear in the list of printers in the Print and Scan system preference and in the list of available printers when a user prints a document. It is also the name of the PPD (Postscript Printer Description) file that the CUPS facility creates for each printer that is added to your Printer preferences.

   Type an optional description in **Location** to assist users in locating the printer.

6. In the **Print Using** window, specify the type of the printer, which enables you to properly manage the printer.

   For example, if you have drivers installed for the printer, click **Select Printer Software** and select the appropriate item such as **HP Laserjet 4300**, then click **OK**.

   You can also specify **Generic Postscript Printer**, or click **Other** to browse for drivers or printer software.

   Click the **Add** button to add the printer to the list of available printers.

7. Repeat this procedure for as many printers as you want to make available for authenticated printing.

You can now use the Copy Files group policy to copy the new *printerName*.ppd file and updated CUPS configuration file (printers.conf) to the appropriate locations on each of your Mac computers in the domain.

To copy printer files to other computers:

1. In the Finder on the Mac template computer, navigate to the /etc/cups directory by clicking **Go > Go to Folder**, then type /etc/cups and click **Go**.
2. Select printers.conf and copy it to the desktop. When prompted, enter your administrator password to copy the file.
3. Open the ppd folder (/etc/cups/ppd). Select the files for all the authenticated printers you defined in the previous procedure and copy them to the desktop.
4. On the desktop, change the file permissions for the printers.conf and *.ppd files so you can copy them to sysvol:
   1. Select the files and click **File > Get Info**.
   2. For each open dialog box, expand **Sharing & Permissions**, then click the lock icon and provide administrator credentials for making changes. Set the permissions for **everyone** to **Read only**.
   3. Reset the lock and close all the open dialogs.
5. On the Windows domain controller create a sub-directory for the printer file in SYSVOL.

   SYSVOL is a well-known shared directory on the domain controller that stores server copies of public files that must be shared throughout the domain. You can use it to copy the printer definition and configuration files to all Mac computers that join the domain.

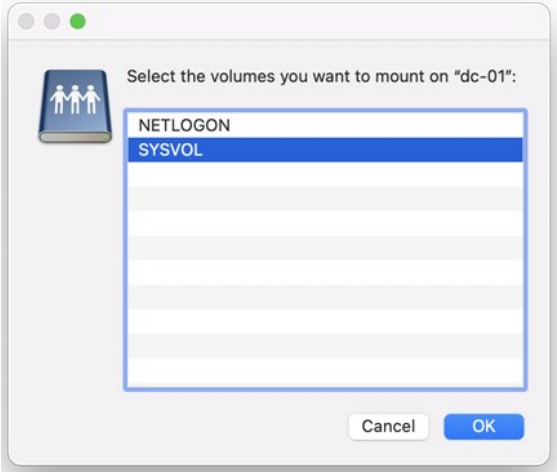
   SYSVOL is located at:

   ```
   C:WindowsSYSVOLsysvoldomainName
   ```

   For example, assuming the domain is acme.com, and using the name MacPrinters for the directory, create the following directory:

   ```
   C:WindowsSYSVOLsysvolacme.comMacPrinters
   ```

6. On the Mac computer, copy the files from the desktop to SYSVOL on the Windows domain controller. If you are connected to the domain, you should see the domain controller in the Finder. If the domain controller is not visible in the Finder, connect to it:
   1. Click **Go > Connect to Server** and select the domain controller.
   2. When prompted select SYSVOL; for example:



3. Navigate to the MacPrinters directory you created, for example by clicking **acme.com** then **MacPrinters**.
4. Drag the printer files to MacPrinters.

7. Configure the Copy Files group policy.
    1. On the Windows domain controller, open the Group Policy Management Editor and select the GPO that is used to manage Mac computers.
    2. Navigate to **Computer Configuration > Policies > Common UNIX Settings** and double-click **Copy Files**.
    3. In **Copy file policy setting**, select **Enabled**.
    4. Click **Add**, then **Browse**. Double-click to open the directory you created for the printer files in Step 5 (for example, MacPrinters).
    5. Select the printers.conf file. Filename now shows MacPrinters/printers.conf.
    6. In **Destination**, type /etc/cups. This group policy will copy printers.conf to the /etc/cups directory of each computer that joins the domain.
    7. Select **Use destination file ownership and permissions**. The file will be assigned the default ownership and permissions:

        owner: root (0)

        group: lp (26)

        permission 0600 (rw- --- ---)
    8. Select **OK** to add the printers.conf file.
8. Click **Add** again and browse to MacPrinters to add the PPD files.
    1. Select one of the PPD files you copied to the MacPrinters directory.
    2. In **Destination**, type /etc/cups/ppd.
    3. Select **Use destination file ownership and permissions**. The file will be assigned the default ownership and permissions:
        owner: root (0)

        group: lp (26)

        permission 0644 (rw- r-- r--)
    4. Click **OK** to add the file.
9. Repeat the sub-steps in Step 8 for each of the PPD files that you have defined, then click **OK** to enable the policy.

    This group policy will copy each *printerName*.ppd file to the /etc/cups/ppd directory of every computer to which the policy applies and that is joined to the domain.

10. Run the adgpupdate command on each target Mac computer to trigger an update of group policies and execute the new Copy Files policy.

    By default, group policies are updated automatically every 90 minutes, so you can skip this step and wait for the automatic update if you wish. You should also log out and back in again on each computer to update the printer configuration dialogs.

Removing a Printer Definition from Client Computers

This section explains how to remove printer definitions that you created for Mac computers in the domain. It assumes that you set up the Copy Files group policy to add printer definitions to each of your joined Mac computers, as explained in Setting up Authenticated Printing.

To remove a printer definition from computers in a domain:

1. Identify the name of the PPD file to delete in /etc/cups/ppd; for example, laserjet4300.ppd.
2. On the Mac template computer (the computer on which you originally defined the authenticated printer), open **System Preferences > Print & Scan**. Select the printer to delete, click the minus (-) button, then click **Delete Printer**.

   Deleting the printer removes the printer from the list, updates the /etc/cups/printers.conf file by removing the definition of the deleted printer, and removes the *printerName*.ppd file from the /etc/cups/ppd directory.
3. Copy the updated printers.conf file to the desktop and change the permissions to **everyone: Read only**.
4. Copy the updated printers.conf file to the SYSVOL and replace the existing file; also remove the PPD file for the deleted printer.

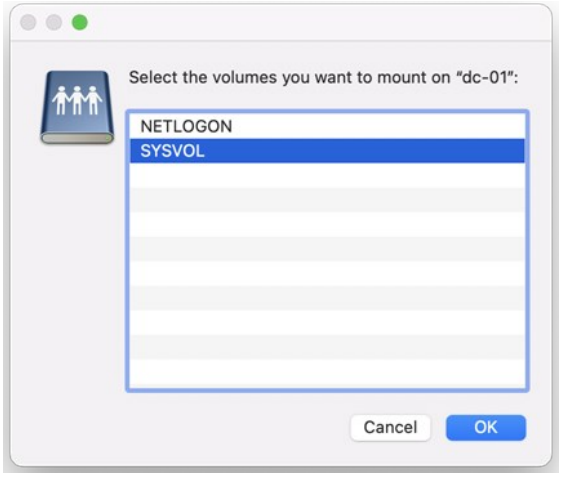
   SYSVOL is a well-known shared directory on the domain controller that stores server copies of public files that must be shared throughout the domain. When authenticated printing was set up, the CUPS configuration file, printers.conf was placed in the SYSVOL/acme.com/MacPrinters folder.

   SYSVOL is located at:

   C:WindowsSYSVOLsysvoldomainName

   If you are connected to the domain, you should see the domain controller in the Finder. If the domain controller is not visible in the Finder, connect to it:

   1. Click **Go > Connect to Server** and select the domain controller.
   2. When prompted, select **SYSVOL**; for example:

   3. Navigate to the directory you created (*domainName*/*subdirectory*), for example by clicking **acme.com** then **MacPrinters**.
   4. Drag the printer configuration file to this directory.
   5. Remove the PPD file for the deleted printer.
5. Remove the deleted printerName.ppd file from the Copy Files policy.
   1. On the Windows domain controller, open the group policy editor and select the policy to edit, such as **Default Domain Policy**.

2. Navigate to **Computer Configuration > Policies > Common UNIX Settings** and double-click **Copy Files**.
3. Select the file to delete and click **Remove**.
4. Click **OK** to save the updated policy.

6. Configure the **Specify commands to run** group policy to remove the deleted *printerName*.ppd file from all the Mac computers in the domain.

    1. In the same folder of the group policy editor (Common UNIX Settings), open the Specify commands to run policy and select **Enabled**.
    2. Click **Add**.
    3. In **Run command**, enter a command similar to the following to remove the *printerName*.ppd file from the /etc/cups/ppd directory on each computer:

        ```
        rm /etc/cups/ppd/*printerName*.ppd; for example:

        rm /etc/cups/ppd/laserjet4300.ppd
        ```

    4. Click **OK** to save the policy.

The next time group policy is updated on computers in the domain (every 90 minutes by default), the following occurs:

- The Copy Files group policy copies the updated printers.conf file to each computer.
- The Specify commands to run group policy removes the specified PPD file on each computer.

Setting Up Local and Remote Administrative Privileges

Delinea provides two group policies to set administrative privileges on the local computer

- Map zone groups to local admin groups allows you to specify one or more zone groups to map to the local admin group. Members of the specified group are given administrative privileges on Mac computers managed by Access Manager.
- Enable administrator access groups allows users in the zone group *ard_admin* to access a computer via Apple Remote Desktop with full privileges.

This section shows you how to use these policies together to enable local and remote administrative access to Mac computers.

To enable remote and local access for a group:

1. Create an Active Directory group, for example, *My_Mac_Admins*, and add users who you want to have administrative privileges.
2. Create an Active Directory group that is a Domain Local Security group. For convenience, name it *ard_admin*.
3. Add *My_Mac_Admins* as a member of *ard_admin*.
4. Create a Delinea zone group, *My_Mac_Admins* and map it to the Active Directory group *My_Mac_Admins*.

    Note: If the local computer is connected to the domain through Auto Zone, you cannot create a zone group because there are no zones. However, all Active Directory groups are valid for the joined computer, so you can map any group, such as *My_Mac_Admins*, to the local admin group, but you need to know the group's UNIX name, which you can retrieve on the local computer, by using the adquery command, as follows

    ```
    [root]#adquery group -n
    ```

    For example, the following shows an adquery command and the name it returns:

    ```
    [root]#adquery group -n |grep -i Mac_Admins   my_mac_admins
    ```

5. Create a zone group, *ard_admin*, and map it to the Active Directory group *ard_admin*.

    Note: This zone group must be named *ard_admin*.

6. In the Group Policy Editor, edit the group policy for the domain, then click **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Accounts > Map zone groups to local admin group**.
7. Open the policy, select **Enable**, then click **Add**. Enter *My_Mac_Admins* (or the name retrieved from the adquery -n command in Step 4), then click **OK**.

    This step maps *My_Mac_Admins* to the admin group on the local computer and gives members of *My_Mac_Admins* all privileges.

8. Click **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Remote Management > Enable administrator**

access groups.

9. Open the policy and select **Enable**.

   This step allows members of *ard_admin* to access a computer via Apple Remote Desktop with full privileges. In Step 7, you effectively gave members of *My_Mac_Admins* administrative privileges. Since *My_Mac_Admins* includes members of *ard_admin*, members of *ard_admin* now have full local and remote administrative access.

Querying User Information for Active Directory Users

When you run commands or use applications that look up user information in the directory, the local Mac directory service is always consulted first before the look-up request is made to Active Directory. If a local user exists with the same name as a UNIX profile name that has been defined for the zone, a lookup request such as id *username* will return the UID and GID associated with the local user account from the local directory service rather than the information associated with the UNIX profile defined in Active Directory.

For example, if you have a UNIX profile in Active Directory for the user mia with the UID of 10024 and the user's primary group is mia with the GID of 10024 and the user is also a member of the Active Directory group users and GID of 10001, running the id mia command returns the following information from Active Directory:

```
uid=10024(mia) gid=10024(mia) groups=10024(mia), 10001(users)
```

However, if there is also a local user account with the same user name of mia, but with a UID of 502 and a primary group named *mia* with a GID of 502, running id mia returns the information for the local user retrieved from the Mac directory service, then any additional group membership information retrieved from Active Directory. For example:

```
id mia

uid=502(mia) gid=502(mia) groups=502(mai), 10001(users)
```

Because the Mac directory service is queried first, the information for the local user mia takes precedence over the information defined in Active Directory. To avoid retrieving the information for a local user instead of the UNIX profile defined in Active Directory, you should make sure that the UNIX profile user names in Active Directory are different from the local user or disable local user accounts.

Migrating from Open Directory to Active Directory

If you install the Delinea DirectControl Agent for Mac in an environment where existing Mac users and computers are managed with Open Directory, you may need to migrate the account information and home directories for those users from the Open Directory environment to Delinea Active Directory. Open Directory and Active Directory support three types of users:

- Local users
- Network home users
- Portable home, or mobile home, users

For example, you may need to migrate existing mobile user accounts from Open Directory to Active Directory or migrate local home directories to a network share.

To migrate users with existing mobile accounts from Open Directory to Active Directory:

1. Create a copy of the user's local home directory in a temporary location if you have enough disk space to do so. This copy can serve as a backup to restore the user's home directory if you run into any synchronization problems.

2. Log on to the Mac client as an administrator.

3. Disable the LDAP service.

   Open the Directory Utility and select the **Services** tab; then deselect **LDAPv3** and click **Apply**.

4. Open a Terminal window and run the following Directory Service command to delete the user's record:

   ```
   dscl /Local/Default -delete /Users/userName
   ```

   where *userName* is a local user; for example, to delete the record for cain:

   ```
   dscl /Local/Default -delete /Users/cain
   ```

5. Navigate to the /Users/user_name/Library/Mirrors directory and delete this folder.

6. Join the Mac computer to an Active Directory domain and restart the computer to shut down and restart services.
7. Create an Active Directory user account for the Open Directory user account, if one does not already exist.

   If you are creating a new Active Directory user, use Active Directory Users and Computers to add the user account.
8. Add the Active Directory user to the Mac computer's zone and define the Delinea Profile for the user:
   - Use the same user name, UID, and GID as the Open Directory user account. You can change this information later with the adfixid program, but for migration you must use the same values.
   - Set the home directory for the user to the appropriate network share using the /SMB/share/path or /AFP/share/path syntax. For example, /SMB/cain/server2003.myDomain.com/Users/cain.

     Note: For synchronizing new mobile user accounts, the empty home directory must exist on the network share. If the user home directories are on the same network share as you previously used with Open Directory, logging on with the new Active Directory account should not affect the files available on the share.

     Because GID values of 0 to 99 are usually reserved for system accounts, you may see a warning message when you save the user's profile if the user's primary GID value is less than 99.

If you have Open Directory users that do not have mobile accounts or portable home directories and you want to synchronize their local home directories with their network home, you should first use the Workgroup Manager to create mobile accounts for those users to establish a portable home directory. You can then follow the steps above to synchronize the portable home directories with their network home directory. If you don't want to synchronize the local home directory with the home directory on the network share, you can simply create Active Directory accounts for the Open Directory users and remove the local user records; see Mapping Local User Accounts to Active Directory for information about removing local user records.

Changing the Delinea UIDs and GIDs

To change the UID and GID values in Delinea Active Directory to match the existing values:

1. Log in to the Mac computer as a local administrator.
2. Open a terminal session.
3. Open the user's home folder and type:

   ```
   ls -ln total 32
   -rw-r--r--@ 1 505 505 3 Mar 26 2007 .CFUserTextEncoding
   -rw-r--r--@ 1 505 505 6148 Mar 26 2007 .DS_Store
   -rw------- 1 505 505 74 Mar 26 2007 .bash_history
   drwx------@ 3 505 505 102 Mar 26 2007 Desktop
   drwx------@ 3 505 505 102 Mar 26 2007 Documents
   drwx------@ 19 505 505 646 Mar 26 2007 Library
   drwx------@ 3 505 505 102 Mar 26 2007 Movies
   drwx------@ 3 505 505 102 Mar 26 2007 Music
   drwx------@ 4 505 505 136 Mar 26 2007 Pictures
   drwxr-xr-x@ 4 505 505 136 Mar 26 2007 Public
   drwxr-xr-x@ 5 505 505 170 Mar 26 2007 Sites
   ```

   The third column shows the UID (505 in this example) and the fourth column shows the GID (also 505).
4. On the Windows workstation, open the Access Manager console. Expand the zone, expand users, and double-click the user to open the property page.
5. Type 505 for the UID.
6. To change the GID, you need to either change the GID of the group to which the user belongs (which will change for all users who belong to that group) or create a new group. To create a new group:

- Open ADUC. Then right-click **Users > New > Group**. Enter a name for the group and click **OK**.
- In the Access Manager console, right click **Groups > Create UNIX Group**. Search for the group you created. Change the GID to the desired value (for example, 505) and click **OK**.

7. To change the GID of the existing group to which the user belongs, expand **Groups** and double-click the group name. Change the GID to the desired value (for example, 505). Click **Yes** on the warning message.

Modifying the Mac UID and GID to Match AD

To change the existing UID and GID to match the values in Active Directory depends on whether you have a local home directory, a network home directory, or a mobile home directory.

To change the existing UID and GID if you have a local home or network home directory:

1. Log in to the Mac computer as a local administrator.
2. Open **Applications > Directory Utility > Services**. Double-click **Active Directory**, then click **Unbind**. Enter your administrator name and password if necessary.
3. Use the ADJoin tool (either the GUI or the command-line version) to connect to an Active Directory domain.
4. Open a terminal session and type the following:

   ```
   id userName
   ```

   Note the primary group. For example:

   ```
   id cain

   ...

   gid=10000(support)
   ```

5. Type:

   ```
   chown -R userName:primaryGroupName /Users/userName
   ```

   For example, for a local home directory:

   ```
   chown -R cain:support /Users/cain
   ```

   For example, for a network home directory:

   ```
   chown -R cain:support /SMB/Users/cain
   ```

To change the existing UID and GID if you have a mobile home directory:

1. Be certain the local home directory is synchronized with the network home directory.
2. Log in to the Mac computer as a local administrator.
3. Open **Applications > Directory Utility > Services**. Double-click **Active Directory**, then click **Unbind**. Enter your administrator name and password if necessary.
4. Use the ADJoin tool (either the GUI or the command-line version) to connect to an Active Directory domain.
5. Open a terminal session and type the following Directory Service command to delete the cached local user:

   ```
   dscl . -delete /Users/userName
   ```

   For example:

   ```
   dscl . -delete /Users/cain
   ```

6. Then type the following commands to remove the home directory so that it syncs again from the network and remove the local copy of mcx so you are prompted to create a mobile account:

   ```
   rm -rf /Users/userName
   ```

```
rm -rf /Library/Managed Preferences/userName
```

7. On the Windows Active Directory computer, set the **User Configuration > Policies > Centrify Settings > Macintosh Settings > Mobility Synchronization Settings** group policies.

   Note: Mobile home directory synchronization is no longer supported since macOS 10.12.

Converting a Local User to an Active Directory User

Although local user accounts can co-exist with Active Directory user accounts, in some cases, you may want to convert some or all of your local accounts to Active Directory user accounts. Converting local users to Active Directory users simplifies account management, but requires you to take some steps manually.

On Mac computers, the local account database is always checked for authentication before Active Directory. If a local user has the same username as an Active Directory user, the local user account is used for authentication. If the local user's password is different from the Active Directory user's password whether logging on using the Mac login window, or remotely (for example, using telnet or ssh), the local user password is required for authentication to succeed. Although authentication succeeds, Access Manager will generate a username conflict warning.

In most cases, you should remove or convert local user accounts to avoid conflicts between Active Directory and local user accounts and to ensure Active Directory password and configuration policies are enforced. If you need to keep local user accounts, you should ensure the logins are distinguishable from Active Directory accounts. For more information, see the Planning and Deployment Guide.

To convert a local Mac user to an Active Directory user:

1. Open a Terminal window and run the following Directory Service command to delete the user's record:

   ```
   dscl /Local/Default -delete /Users/userName
   ```

   where *userName* is a local user; for example, to delete the record for cain:

   ```
   dscl /Local/Default -delete /Users/cain
   ```

   Although the user record is deleted, the home directory for the user (/Users/cain), including all sub-directories and files, still exists. When you create an Active Directory user with the same name, this user will have access to everything in the existing local home directory.

2. On a Windows computer, use Active Directory Users and Computers to create an Active Directory user account for the local user account (for example, cain), if one does not already exist.

3. In the Access Manager console add the Active Directory user to the appropriate zone and define the Delinea Profile for the user. Set the home directory for the user:

   Note: The default home directory for Mac users is the /Users directory, unlike most UNIX systems where /home is the default by convention.

   - To a local home directory: /Users/*userName*; for example, /Users/cain.
   - To an appropriate network share using the /SMB/share/path or /AFP/share/path syntax. For example, /SMB/cain/server2003.myDomain.com/Users/cain. See Configuring a network home directory.
   - To a network home directory. If you wish to create a mobile account for the user and synchronize the user's folders the next time the user logs on, see Configuring a Portable Home Directory.

4. Reboot the Mac computer, then log in as the new Active Directory user.

Migrating a User from Apple's Active Directory Plugin to Delinea Active Directory

When you create an Active Directory user by using the Mac Directory Utility Active Directory plug-in it creates numeric user (UID) and group (GID) identifiers. When you migrate a current Active Directory user to Delinea Management Services for Mac, the Access Manager console creates a UID and GID that are different than the current UID and GID. When an Active Directory user attempts to log in after the agent is installed, the changed UID and GID cause ownership and permission problems with the user's home directory.

There are two basic approaches to solving this problem:

- Changing the Delinea UIDs and GIDs
- Modifying the Mac UID and GID to Match AD

Using Apple's Scheme to Generate UIDs And GIDs For Mac Users

By default, Delinea uses a different scheme than the Apple Active Directory plugin to generate numeric user (UID) and group (GID) identifiers for Mac users added to Active Directory. If you use the default Delinea scheme to generate identifiers, you must resolve UID and GID conflicts after migrating users. For example, after migrating you can change ownership on the existing files (see Modifying the Mac UID and GID to Match AD otherwise users have Delinea-generated UIDs whereas their files belong to Apple-generated UIDs so users will be unable to access files and folders in their home directories.

On the other hand, Delinea allows you to use the Apple scheme, rather than the default Delinea scheme, to create UIDs and GIDs for migrated users. This method ensures compatibility with Mac tools, such as ExtremeZ-IP, that require UIDs and GIDs generated with the Apple scheme, not the Delinea scheme.

This section explains how to create Apple-generated UIDs and GIDs for Mac users who you are adding to Active Directory with Delinea Management Services for Mac when a computer is connected to Delinea Active Directory through Auto Zone.

> **Note**: If your computer is joined to a zone, however you are adding users to the zone, you can choose to use the Apple scheme to generate UID and GID values. For example, you can specify the Apple scheme with adedit, with the Zone Provisioning Agent, and in the Access Manager Console.

Delinea provides the auto.schema.apple_scheme parameter to enable use of the Apple schema for generating UIDs for new users. The recommended way to set this parameter is by way of group policy so that you can set it for a group of computers. You may also set the parameter on individual computers by editing the Delinea configuration file

To use group policy to enable the Apple scheme for generating UIDs and GIDs:

1. If you are generating new UIDs and GIDs for files that reside remotely in AFP or NFS mounted shares, back up the UIDs and GIDs on the computer where the share resides by executing a command similar to the following:

   ```
   adquery user > olduid
   ```

   > **Note**: You do not need to perform this step for Samba shares.

2. On a Windows computer, open the Group Policy Management Editor and edit a group policy object that applies to Mac computers.
3. Expand **Computer Configuration › Policies › User Configuration › Policies › Centrify Settings › Direct Control Settings › Adclient Settings**, and double-click **Generate New UID/GID using Apple scheme in Auto Zone**.
4. Select **Enabled** and click **OK** to set the policy.

To edit the configuration file and enable the Apple scheme for generating UIDs and GIDs on a single computer:

1. If you are generating new UIDs and GIDs for files that reside remotely in AFP or NFS mounted shares, back up the UIDs and GIDs on the server where the computer resides by executing a command similar to the following:

   ```
   adquery user > olduid
   ```

   > **Note**: You do not need to perform this step for Samba shares.

2. Log in to a Mac computer.
3. Edit the Delinea configuration file: /etc/centrifydc/centrifydc.conf.
4. Find the following parameter, remove the comment and set its value to true:

   ```
   auto.schema.apple_scheme: true
   ```

You may also enable the Apple scheme to set the primary GID for users if you wish.

> **Note**: You may set the primary GID in this way only if the parameter auto.schema.private.group is set to false. Otherwise, the primary GID is set to the value of the user's UID.

To enable the Apple scheme for generating the primary GID:

1. If you are generating new UIDs and GIDs for files that reside remotely in AFP or NFS mounted shares, back up the UIDs and GIDs on the computer where the share resides by executing a command similar to the following:

   ```
   adquery user > olduid
   ```

   > **Note**: You do not need to perform this step for Samba shares.

2. In the Group Policy Management Editor, edit a group policy object that applies to Mac computers, expand **Computer Configuration › Policies ›**

User Configuration > Policies > Centrify Settings > Direct Control Settings > Adclient Settings, and double-click **Set user's primary gid in Auto Zone**.

3. Select **Enabled**.
4. In **Set user's primary gid in Auto Zone**, type -1.

   The primary GID for each user will be generated by the Apple scheme, as specified with the "Generate New uid/gid using Apple scheme in Auto Zone" group policy, which you enabled in the previous procedure.
5. Click **OK** to save the setting.

After setting these policies, run adgpupdate to update the group policies you just set, and flush the cache on each joined computer to update the UID and GID values for any existing users.

To flush the cache on each Mac computer:

1. Log in to a Mac computer and open the Terminal application.
2. Execute the following command as root:

   adflush

New users who you migrate to Active Directory from the Apple Active Directory plug-in will automatically keep the same UID, GID, and primary GID values that they had before migration, and their home ownership will work properly.

After you flush the cache, any existing users and groups will have their UID, GID, and primary GID values changed from the Delinea scheme to the Apple scheme. However, ownership of files and folders in home directories will still belong to the Delinea UID and GID. To change ownership to the new UID and GID generated by the Apple scheme, run the fixhome.pl script as explained in the following procedure.

To correct file ownership by running fixhome.pl

Note: If you generated new UIDs and GIDs for files that reside remotely in AFP or NFS mounted shares, the fixhome.pl script does not have permission to change UIDs and GIDs in the share, and you must manually update the UIDs and GIDs on the server where the share folders reside. In this scenario, skip to *Workaround for AFP and NFS Mounted Shares* below and continue from there.

For Samba shares and local UIDs and GIDs:

1. Log in on a Mac computer for which you have changed UID and GID values to the Apple scheme.
2. Execute the following command as root:

   /usr/local/share/centrifydc/sbin/fixhome.pl

The script changes ownership of files and folders in the home directory of all Active Directory users from the Delinea-generated UID or GID to the Apple-generated UID or GID.

The script uses /Users as the root for all home directories. You may specify a different home root if necessary by using the --dir option. Use the --help option to see a list of options that you can specify with this command:

```
/usr/local/share/centrifydc/sbin/fixhome.pl --help
```

For example, you can run the command in test mode to see the changes that will be made, but without committing the changes:

```
[root]#/usr/local/share/centrifydc/sbin/fixhome.pl --test

User Home UID Map GID Map

user1 /Users/user1 796918879=>558948313 20=>5287576209
```

Or you could update specific users rather than all users:

```
[root]#/usr/local/share/centrifydc/sbin/fixhome.pl --include user2
```

Workaround for AFP and NFS Mounted Shares

For AFP and NFS mounted share folders (or remote file systems), fixhome.pl does not have permission to change the UID/GID of files in the folder. Perform the

following steps to work around this issue:

1. On the server where the share folders reside, open the UID/GID backup file to have access to the old UID/GID strings.
2. On the server where the share folders reside, change the old UIDs and GIDs to the new UIDs and GIDs one at a time by executing commands similar to the following:

```
find ShareFolder -user previous_uid -group previous_gid -exec chown new_uid:new_gid {} ;
```

To enable the Apple scheme for mobile users:

Additional steps are required to enable the Apple scheme for mobile users. After enabling the Apple scheme as described in the preceding sections, you must ensure that the UID and PGID for the mobile user's local user record match the UID and PGID used by the DirectControl agent.

1. Change the UID and PGID in the local user record so that they match the IDs used by the agent:
    1. Open **Users and Groups**.
    2. Right-click the mobile user account.
    3. Choose **Advanced Options**, and change the UID and PGID so that they match the IDs used by the agent.
2. After changing the UIDs and PGIDs of mobile users, run the fixhome.pl script as described above in *To correct file ownership by running fixhome.pl*.

To use the Zone Provisioning Agent to enable the Apple scheme for generating UIDs and GIDs:

1. Ensure that the Zone Provisioning Agent is configured as described in the section "Configure the Zone Provisioning Agent" in the *Planning and Deployment Guide*.
2. Ensure that zone provisioning groups are created and configured as described in Chapter 8, "Preparing the Environment for Migration of Existing Users and Groups" in the *Planning and Deployment Guide*.
3. Start Access Manager.
4. In the console tree, expand the **Zones** node.
5. Select the top-level parent zone, right-click, then click **Properties**.
6. Click the **Provisioning** tab.
7. Click **Enable auto-provisioning for group profiles**.
8. Click the Find icon to search for and select the primary group (typically the Domain Users group) as the Source Group.
9. Select **Generate using Apple scheme** as the method for assigning a new GID to new UNIX group profiles.

    This method generates group GIDs based on the Apple algorithm for generating numeric identifiers from the Active Directory group's objectGuid. This option is only supported for hierarchical zones.
10. Select a method for assigning a new group name to new UNIX group profiles:
    - **SamAccountName attribute** generates the group name for the new UNIX group profile based on the samAccountName value.
    - **CN attribute** can be used if you verify the common name does not contain spaces or special characters. Otherwise, you should not use this option.
    - **RFC 2307 attribute** can be used if you have added the RFC 2307 groupName attribute to Active Directory group principals. Otherwise, you should not use this option.
    - **Zone default value** to use the setting from the Group Defaults tab for the zone. In most cases, the default is a variable that uses the sAMAccountName attribute.
    - By default, all UNIX group names are lowercase and invalid characters are replaced with underscores.
11. Click **OK** to save your changes.

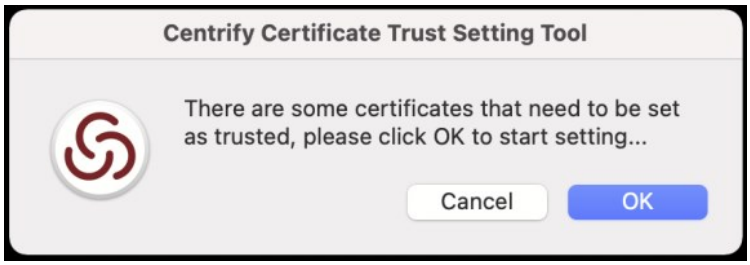
Configuring Auto-Enrollment

Delinea uses the Microsoft Windows certificate auto-enrollment feature to make certificates available to UNIX and Mac computers. If auto-enrollment is enabled, when a UNIX or Mac computer joins a domain, certificates are requested from the Certification Authority based on particular templates, and the certificates are installed on the joined computer

To enable auto-enrollment:

- Enable auto-enrollment for the group policy.
- Create a certificate template with auto-enrollment enabled.

> **Note**: As of MacOS Big Sur (11.0), Apple no longer allows silently adding root certificates to Keychain with a trusted setting. If there are some root certificates installed from your domain by the Delinea Agent, the Delinea Certificate Trust Setting Tool will open automatically. Please follow the instructions to set certificates as trusted.



Configuring 802.1X Wireless Authentication

This section explains how to configure Active Directory and Mac to authenticate Active Directory users by using a Microsoft RADIUS server with the 802.1X PEAP (MSCHAPv2) protocol over a wireless network from a Mac computer.

On Mac OS X, 802.1X wireless authentication does not rely on Delinea Access Manager or Apple's Active Directory plugin but is configured primarily through group policies that apply to the Windows server and the Mac computers.

System Configuration for 802.1X Wireless Authentication

The following table summarizes the environment that is needed for 802.1X wireless authentication:

Windows side	Windows Server 2003 R2 Enterprise Edition Domain Controller (supports PEAP) with Internet Authentication Service (IAS) installed; on Windows server 2003, RADIUS server is part of IAS. or Windows Server 2008 R2 Enterprise Edition Domain Controller (supports PEAP/TLS) with Network Policy Server (NPS) installed; on Windows Server 2008, Radius server is part of NPS.
	Active Directory on the Windows Server
	Group Policy Management Console (GPMC), which is required to configure 802.1x group policies and deploy certificates.
	Certificate Services, which is required to obtain the required certificates.
	Access Manager console 5.1.*x* or later, which is required to set group policies that apply to Mac computer.
Mac side	DirectControl agent 5.0.1-171 or later to enforce group policies on the Mac computer.
Wireless access point device	Supports 802.1x wireless authentication through one of these protocols: * WPA Enterprise WPA2 Enterprise 802.1X WEP (the name can be different, for example, RADIUS)

> **Note**: Although it is possible to configure other RADIUS servers for 802.1X wireless authentication, or use other protocols, this document focuses on the Microsoft RADIUS server and the PEAP and TLS protocols.

These instructions assume that you have a RADIUS server properly configured for 802.1X wireless authentication, so that you can now proceed to configure your Mac environment. For a description of how the RADIUS server must be configured to support 802.1X wireless authentication on Mac OS X, see the section below named *Confirming that Windows Server Supports Certificate Auto-enrollment*. Click a link if you have questions about whether your RADIUS server is configured properly with regard to any particular item:

Of course, there are other configuration steps that are required to set up a RADIUS server, such as configuring the RADIUS client and configuring a remote access policy, however, the important consideration for Mac 802.1X authentication is that the specified certificate and private key have been created and deployed to the domain. When a Mac computer joins a Windows domain, Access Manager automatically finds certificates on the Domain Controller and adds them as trusted certificates to Keychain Access on the Mac computer.

Once you are certain that the RADIUS server is properly configured, you can configure your Mac environment; see the following section for instructions on configuring OS X 10.7 or later.

Configuring Mac OS X 10.7 or Later for 802.1X Wireless Authentication

Mac OS X 10.7 changed the way to create and manage profiles such that configuring 802.1X wireless authentication varies significantly between 10.7 and earlier versions of OS X. This section explains how to configure a Mac OS X 10.7 or later computer for 802.1X wireless authentication.

Before configuring your Mac environment, be certain that the RADIUS server is configured as described above in the section, *System Configuration for 802.1X Wireless Authentication*. This configuration includes a domain root CA certificate or RAS/IAS server certificate, as well as a private key that are required to be trusted on the Mac computer.

However, there are no manual steps that you must perform to trust these certificates on your Mac computers. As mentioned previously, when a computer is joined to a domain, Access Manager automatically looks for certificates on the domain controller, and adds these certificates and the private key to the system Keychain on the Mac computer.

Through group policy settings you can use these certificates to create two different types of system profiles

- To create a system profile that allows users to authenticate to an 802.1X-protected ethernet network, see the next procedure, *To configure Mac OS X 10.7 or Later to Create an 802.1X Ethernet Profile*.
- To create a system profile that allows users to authenticate to an 802.1X wireless network, see the procedure further down, *To configure Mac OS X 10.7 or Later to Create an 802.1X WiFi Profile*.

The certificate template — as well as a certificate chain file and private key — are pushed to `/var/Centrify/net/certs` on the Mac computer when it joins the domain. Before you configure the group policy for the Mac computer, if you want to verify that auto-enrollment is operating correctly, you can open a Terminal window on the Mac computer and run a command similar to the following to check that the certificate has been downloaded to the computer:

```
admin$ls /var/Centrify/net/certs |grep -i auto_
...
auto_TemplateName.cert
auto_TemplateName.chain
auto_TemplateName.key
```

You should see three `auto_` files as shown in the example.

To configure Mac OS X 10.7 or later to create an 802.1X Ethernet profile

1. On a Windows computer, open the Group Policy Management Editor and edit a group policy object that applies to Mac computers.
2. Expand **Computer Configuration > Policies > User Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Settings**, and double-click **Enable Ethernet Profile**.
3. Select **Enable**, then click **Add**.
4. Type the name of the auto-enrollment machine certificate that has been pushed down from the Windows domain server.

   When pushed to a Mac computer, certificate names are prepended with `auto_`; for example:

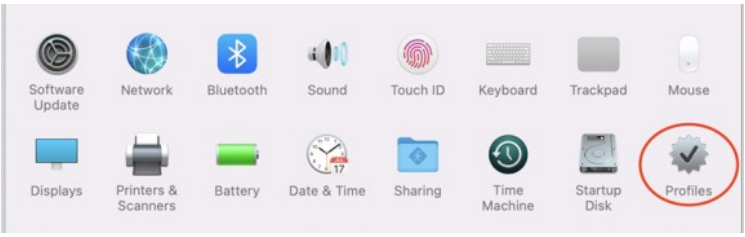
   ```
   auth_Centrify-1X
   ```

   This group policy runs a script that looks for the specified certificate template in the /var/Centrify/net/certs directory (which contains the certificate templates pushed down to Mac when they join the domain) and creates a WiFi profile from this certificate.
5. Click **OK** to save the profile information and **OK** again to save the policy setting.

   Note: This group policy will take effect at the next group policy update interval, or you can run `adgpupdate` in a terminal window on the Mac computer to have the policy take effect immediately.

When the group policy takes effect, it runs a script to create an ethernet profile for the computer from the certificate template and private key downloaded from the domain controller. This policy supports the TLS protocol for certificate-based authentication. The Mac computer is now configured for access to the radius access point.

On the Mac computer you can view the profile in System Preferences.



To configure Mac OS X 10.7 or later to create an 802.1X WiFi profile

1. On a Windows computer, open the Group Policy Management Editor and edit a group policy object that applies to Mac computers.
2. Expand **Computer Configuration › Policies › User Configuration › Policies › Centrify Settings › Mac OS X Settings › 802.1X Settings**, and double-click **Enable Wi-Fi Profile**.
3. Select **Enable**, then click **Add**.
4. Enter the following information for the Wi-Fi profile:

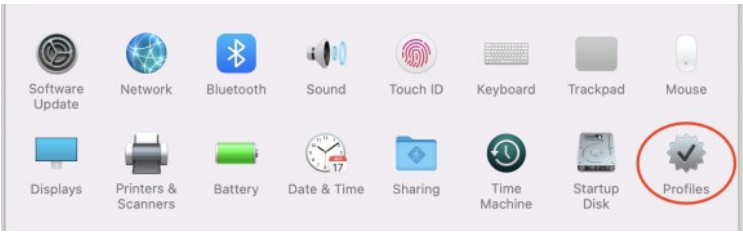
SSID	Type the SSID for the wireless network.
Template name	Type the name of the auto-enrollment machine certificate that has been pushed down from the Windows domain server. When pushed to a Mac computer, certificate names are prepended with auto_; for example: auth_Centrify-1X This group policy runs a script that looks for the specified certificate template in the /var/Centrify/net/certs directory (which contains the certificate templates pushed down from the domain controller) and creates an ethernet profile from this certificate.
Security type	Select the Security type from the drop-down list.
Other options	Select one or more of the following options: **Auto join**: Select this option to specify that the computer automatically join a Wi-Fi network that it recognizes. Do not select this option to specify that the logged in user must manually join a Wi-Fi network. **Hidden network**: Select this option if the Wi-Fi network does not broadcast its SSID.

1. Click **OK** to save the profile information and **OK** again to save the policy setting.

   Note: This group policy will take effect at the next group policy update interval, or you can run adgpupdate in a Terminal window on the Mac computer to have the policy take effect immediately.

When the group policy takes effect, it runs a script to create a WiFi profile for the computer from the certificate template and private key downloaded from the domain controller. This policy supports WEP or WPA/WPA2 security with the TLS protocol for certificate-based authentication. The Mac computer is now configured for access to the radius access point.

On the Mac computer you can view the profile in System Preferences.



Confirming that Windows Server Supports Certificate Auto-enrollment

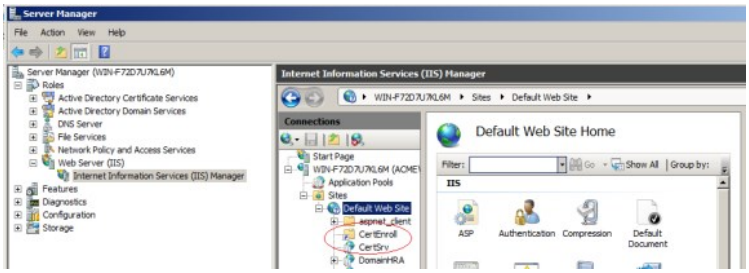
This section describes how the RADIUS server must be configured to support 802.1X wireless configuration for Mac computers.

Internet Information Services (IIS) Supports CertEnroll and CertSrv URLs

IIS must support the CertEnroll and CertSrv URLs to enable web-based access to certificate tasks.

To verify that Iis supports the CertEnroll and CertSrv URLs

1. On the Windows Certificate Authority server, click **Start > Administrative Tools > Server Manager** to open Server Manager.
2. Expand **Roles > Web Server (IIS)** and click **Internet Information Services (IIS) Manager**.
3. In the right, **Connections** pane, expand **Sites > Default Web Site** and you should see CertEnroll and CertSrv:



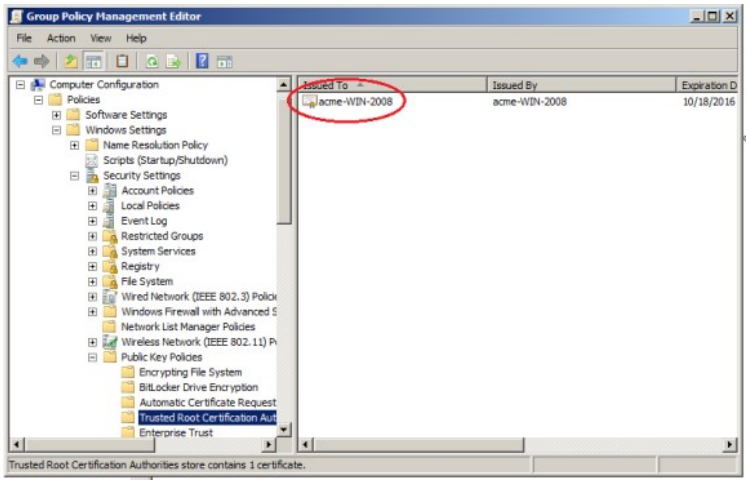
Windows Public Key Group Policies are Set to Trust the Root Certificate Authority and Enroll Certificates Automatically

Through group policy settings, the root certificate must be imported into the Trusted Root Certification Authorities group policy and set to enroll certificates automatically.

To verify that Windows public key group policies are set to trust the root certificate authority and enroll certificates automatically

1. On the Windows Certificate Authority server, click **Start > Administrative Tools > Server Manager** to open the Group Policy Management Editor.
2. Expand **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies** and select **Trusted Root Certification Authorities**.

   You should see your root certificate:



3. Expand **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies** and double-click **Certificate Services Client - Auto-Enrollment**.

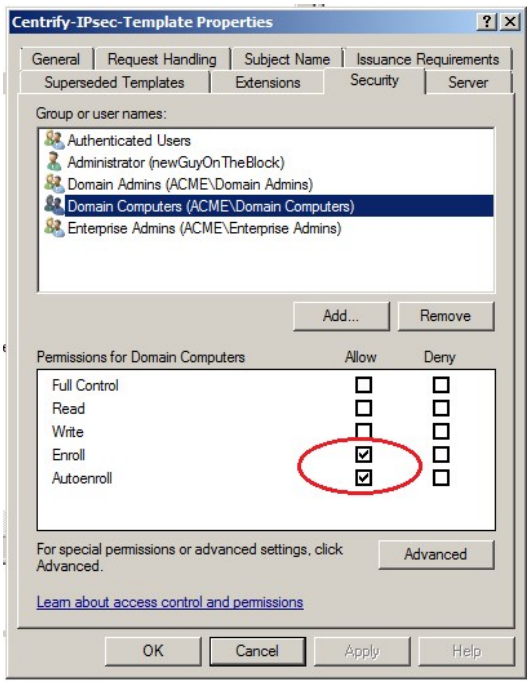
4. In **Configuration Model** select **Enabled**.
5. Select both boxes, **Renew expired certificates** and **Update certificates that use certificate templates**.
6. Click **OK** to save the policy.

A Certificate Template is Configured to Automatically Enroll Domain Computers

To automatically enroll domain computers, you must have a certificate template that supports auto-enrollment for domain computers.

To configure a certificate template to automatically enroll domain computers

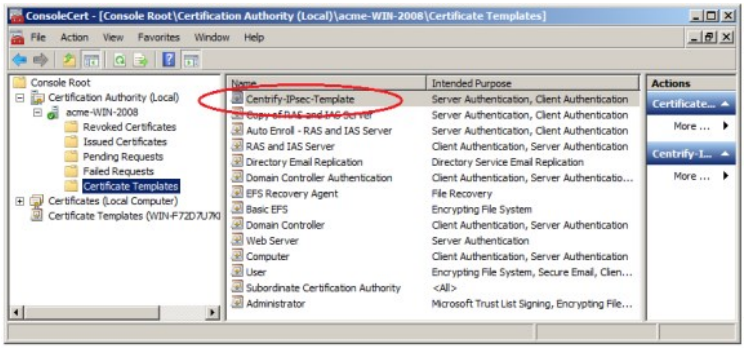
1. On the Windows Certificate Authority server, open an mmc console that contains the Certification Authority and Certificates snap-ins (**Start > Run >** mmc.exe).
2. If snap-ins for Certificate Templates, Certificates, and Certifications Authority are not displayed under Console Root in the navigation pane, add them now. To do so, click **File > Add/Remove Snap-in**.
    1. Select **Certificate Templates** and click **Add**.
    2. Click **Certificates** and click **Add**.
    3. Select **Computer Account** and click **Next**.
    4. Select **Local computer** and click **Finish**.
    5. Select **Certification Authority** and click **Add**.
    6. Select **Local computer** and click **Finish**.
    7. Click **OK**.
3. Select **Certificate Templates (domainController)** in the navigation pane.
4. In **Certificate Templates**, duplicate the Workstation Authentication certificate. Right-click **Workstation Authentication** and select **All Tasks > Duplicate Template**.
5. Perform the following steps in the **Properties of New Template** dialog:
    1. In the **General** tab, type a template name of your choice (for example, **Mac Auto-Enroll Certificates**) in the **Template name** field (do not use special characters such as brackets and asterisks). Type the same name in the **Template display name** field so that the template displays by that name in the Certificate Templates list.
    2. In the **Extensions** tab, select **Application Policies > Edit**. In the resulting dialog, select **Add > Server Authentication** and click **OK**.
    3. In the **Extensions** tab, verify the **Client Authentication** is already in the application policy list. If it is not, add it in the same way that you added the **Server Authentication** policy.
    4. In the **Subject Name** tab, select **Build from this Active Directory information**. In the **Subject name format** field, select **Fully distinguished name**. In the **Include this information in alternate subject name** list, select **User Principle Name (UPN)**.
    5. In the **Security** tab, select **Domain Computers (domainController)** and ensure that the template is enabled for Enroll and Autoenroll.



6. Click **Apply** and **OK** to save your settings.

6. Verify that the new template has been added to the certification authority.

   Expand **Console Root > Certification Authority > domainController** and select **Certificate Templates**. You should see that the certificate template that you have configured for auto-enrollment is contained in the certification authority for the domain:



   If the new certificate template is not contained in the certification authority, add it now:

   1. In the navigation pane, right-click **Certification Templates** under **Console Root > Certification Authority >** domainController.
   2. Select **New > Certificate Template to Issue**.
   3. Scroll to the newly created template, select it, and click **OK**.

7. Enable the following group policy:

   - On Windows 2008: **Computer configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Certificate Services Client - Auto-Enrollment Settings**.
   - On Windows 2012: **Computer configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Certificate Services Client - Auto-Enrollment**

     Note: To enable a group policy, open the Group Policy Management console by selecting **Start > Administrative Tools > Group Policy Management**. In the Group Policy Management console navigation pane, expand **Group Policy Management >**

ForestName > **Domains** > DomainName > **Group Policy Objects**. Right-click **Default Domain Policy** and select **Edit**. In the resulting Group Policy Management Editor, navigate to the group policy described above and double-click the group policy. In the resulting dialog, select **Enabled** in the **Configuration Model** field.

8. On the Mac computer, download the certificates by executing the following commands in a terminal window:

   ```
   sudo adflush
   ```

   ```
   adgpupdate
   ```

9. Verify that the certificates were downloaded:

   1. On the Mac computer, open Keychain Access and verify that the certificates are there.
   2. On the Mac computer, verify that the certificates are in /var/Centrify/net/certs.
   3. On the Windows Certificate Authority server, open the Certification Authority console (**Start** > **Run** > certsrv.msc) and verify that the certificates are in the **Issued Certificates** folder.

A Certificate Template is Configured to Automatically Enroll Domain Users

To automatically enroll domain users, you must have a certificate template that supports auto-enrollment for domain users.

To configure a certificate template to automatically enroll domain users

1. On the Windows Certificate Authority server, open an mmc console that contains the Certification Authority and Certificates snap-ins (**Start** > **Run** > mmc.exe).
2. Verify that the snap-ins described in Step 2 are present under Console Root in the navigation pane. If they are not, add them now as described in Step 2.
3. Select **Certificate Templates (domainController)** in the navigation pane.
4. In **Certificate Templates**, duplicate the User certificate. Right-click **User** and select **All Tasks** > **Duplicate Template**.
5. Perform the following steps in the **Properties of New Template** dialog:
   1. In the **General** tab, type a template name in the **Template name** field. Type the same name in the **Template display name** field so that the template displays by that name in the Certificate Templates list. For Mac, you can specify a name of your choice (do not use special characters such as brackets and asterisks). For mobile devices, the template name *must* be **User-ClientAuth**.
   2. In the **Security** tab, select **Domain Users (domainController)** and ensure that the template is enabled for Enroll and Autoenroll.
   3. Optionally, in the **Subject Name** tab, select **Build from this Active Directory information**. De-select the **Include email in subject name** and **E-mail name** check boxes. If you perform this step, Active Directory users do not need an email address.
6. Verify that the new template has been added to the certification authority as described in Step 6. If the new certificate template is not contained in the certification authority, add it now as described in Step 6.
7. Enable the following group policy:
   - On Windows 2008: **Computer configuration** > **Windows Settings** > **Security Settings** > **Public Key Policies** > **Certificate Services Client - Auto-Enrollment Settings**.
   - On Windows 2012: **Computer configuration** > **Windows Settings** > **Security Settings** > **Public Key Policies** > **Certificate Services Client - Auto-Enrollment**.

     Note: See Step 7 for details about how to enable the group policy.
8. On the Mac computer, download the certificates by executing the following commands in a terminal window.

   As the local Administrator:

   ```
   sudo adflush
   ```

   As an Active Directory user:

   ```
   adgpupdate
   ```

9. Verify that the certificates were downloaded:

1. On the Mac computer, open Keychain Access and verify that the certificates are in the Login keychain.
2. On the Mac computer, verify that the certificates are in ~/.centrify/:

```
ls -l ~./centrify/
```

Configuring Single Sign-On for SSH and Screen Sharing

On OS X 10.10 and later, you can change configuration settings to allow single sign-on for SSH and Screen Sharing using Kerberos. Kerberos authorization for SSH and Screen Sharing allows you to establish an SSH or Screen Sharing connection to configured target machines joined to the same domain within the same single sign-on (SSO) session. In addition to authorizing SSH or Screen Sharing for the currently logged in user, you can authorize SSH or Screen Sharing for a different smart card user (for example, an admin user) by obtaining that user's Kerberos credentials.

To configure SSH SSO

Note: Smart card authentication for SSH sessions across different forests or domains is not supported.

1. Verify that all client and target machines are joined to the same AD domain.

   See **Joining an Active Directory Domain** for more information.

2. Enable GSSAPIAuthentication and GSSAPIDelegateCredentials in the /etc/ssh/ssh_config (/etc/ssh_config on OS X 10.10) file on both the client and target machine.

   ```
   GSSAPIAuthentication yes

   GSSAPIDelegateCredentials yes
   ```

3. Enable GSSAPIAuthentication and GSSAPIKeyExchange in the /etc/ssh/sshd_config (/etc/sshd_config on OS X 10.10) file on both the client and target machine.

   ```
   GSSAPIAuthentication yes

   GSSAPIKeyExchange yes
   ```

   Note: As of macOS 10.12, Apple's built-in ssh server no longer supports as the target machine. You can still use SSH SSO to login to other server machines, such as Linux/UNIX machines.

4. Enable adclient.krb5.autoedit on the target machine.

   The easiest way to do this is enabling the **DirectControl Settings > Kerberos Settings > Manage Kerberos configuration** group policy.

5. Restart Delinea Management Services on the target machine.

   ```
   $ sudo /usr/local/share/centrifydc/bin/centrifydc restart
   ```

   The logged in user can now open SSH connections to the target machine using a FQDN.

   ```
   $ ssh hostname.domainname
   ```

To configure Screen Sharing SSO

Note: Single sign-on for Screen Sharing requires Mac OS X 10.11 or higher.

1. Verify that both the client and target machines are updated to at least Delinea Management Services 5.3.1.

   ```
   $ adinfo -v

   adinfo (CentrifyDC 5.3.1-xxx)
   ```

   If an update is necessary, refer to Upgrading the Delinea DirectControl Agent for Mac for instructions and best practices.

2. Open **System Preferences > Sharing**, then select **Screen Sharing** and specify which users can initiate Screen Sharing sessions in the **Allow access for:** list.

Note: Only Screen Sharing supports SSO, as Remote Management can not allow access for network users.

The logged in user can now open Screen Sharing connections to the target machine using a FQDN.

```
$ open vnc://hostname.domainname
```

To obtain Kerberos credentials for a smart card user for SSH or Screen Sharing SSO

1. Complete all the steps in **To Configure SSH SSO** and **To Configure Screen Sharing SSO** above.
2. Insert the user's smart card into the reader.
3. Obtain Kerberos credentials from the smart card currently in the reader and use those credentials to authorize SSH.

   For multi-user PIV cards or multi-user smart cards:

   ```
   $ /usr/local/bin/sctool -a unixName
   ```

   For all other smart cards:

   ```
   $ /usr/local/bin/sctool -k userPrincipalName
   ```

   Refer to **Understanding sctool** for more information about the sctool `-a` and `-k` options.

   After unlocking the smart card, you can now open SSH or Screen Sharing connections to the target machine using the obtained Kerberos credentials.

Configuring FileVault 2

FileVault 2, available in OS X 10.8 and later, allows encryption of an entire drive to keep data secure. Although you can enable FileVault 2 through System Preferences on your Mac computers, using Delinea Management Services for Mac to configure FileVault 2 through group policy provides the advantage of creating an institutional recovery key for each of your Mac computers. Two different recovery key approaches—institutional and personal—guarantee that you will always have access to all of your encrypted computers, even if users forget their passwords.

For more information about FileVault 2, see the following Apple Knowledge Base article: "OS X: About FileVault 2".

How Filevault2 Protection Is Enabled by Delinea

Delinea relies on two features to enable FileVault 2 protection

- The "Managed By" user setting, which specifies an Active Directory user who can manage and unlock an encrypted disk.

  You specify the "Managed By" user in Active Directory Users and Computers on the domain controller. The "Managed By" user is associated with the Mac computer object, so it is possible for each computer to have its own "Managed By" user.

- The FileVault recovery key, which can be either one "institutional" key that is applied to multiple Mac computers, or computer-specific keys which are generated individually for each Mac computer.
  - If you choose to use one institutional key, you first create a FileVaultMaster certificate, which is applied to Mac computers through the Enable FileVault 2 group policy.

    When you enable the Enable FileVault 2 group policy, the FileVaultMaster certificate is applied to Mac computers automatically at the next scheduled group policy update interval. Or, you can apply the FileVaultMaster certificate immediately by executing the adgpupdate command.

  - If you choose to use computer-specific keys that are unique to each Mac computer, you do not create a FileVaultMaster certificate.

Instead, the key is generated automatically when the “Managed By” user logs into the Mac computer for the first time and then logs out. The key, which is the “Managed By” user’s personal key, is then stored in the computer’s computer object in Active Directory.

>**Note**: Enabling the Enable FileVault 2 group policy does not enable FileVault 2 protection on the Mac computers to which the group policy is applied. Instead, FileVault 2 protection is enabled on Mac computers as described in the remainder of this section.

The following list describes the overall process that results in FileVault 2 protection being enabled on a Mac computer.

1. The “Managed By” user is set in ADUC for one or more Mac computers.
2. The Enable FileVault 2 group policy is enabled.
   - If you select the **Use Institutional Recovery Key** option in the group policy, the FileVaultMaster certificate is applied to Mac computers. In this situation, all of the Mac computers to which the group policy was applied use the same key.
   - If you did not select the **Use Institutional Recovery Key** option in the group policy, a recovery key is not generated until the “Managed By” user logs into a Mac computer.
3. A user logs into a Mac computer. If FileVault 2 protection is not already enabled on the computer, the user’s Active Directory credentials are checked to verify that the user is the “Managed By” user. For this step to complete successfully, one of the following conditions must exist:
   - The Mac computer must be able to communicate with the domain controller (that is, it must be in connected mode), or
   - If the Mac computer is disconnected from the domain controller, locally cached AD user credentials must be available in the Delinea cache.
4. When the user is verified to be the “Managed By” user, one of the following actions takes place:
   - If you selected the **Use Institutional Recovery Key** option in the Enable FileVault 2 group policy, the FileVaultMaster certificate data is used to enable FileVault 2 protection on the computer.
   - If you did not select the **Use Institutional Recovery Key** option in the Enable FileVault 2 group policy, a personal recovery key is created for the computer and stored in the computer object in Active Directory. The personal recovery key is used to enable FileVault2 protection on the computer.

FileVault 2 Configuration Overview

Configuring a Mac computer for FileVault 2 protection requires configuration steps on both the Mac computer and the domain controller (or any Windows computer on which you can configure Group Policy on the domain controller). The following is a list of the major steps in the process, with links to each procedure that you must complete.

1. Create FileVault master keychain. The master keychain contains a private key that can be used to unlock the encrypted disk.

   Note: This step is required only if you are using one institutional key for multiple Mac computers. If you are using computer-specific (“personal”) keys, go to Step 4.

2. Export certificate from FileVault master keychain and upload it to a domain server. Uploading the certificate to a domain server allows you to select it when you enable the “FileVault 2” group policy.

   Note: This step is required only if you are using one institutional key for multiple Mac computers. If you are using computer-specific (“personal”) keys, go to Step 4.

3. Enable BitLocker Recovery Password Viewer in Active Directory.

   This step is required only if you are using computer-specific (“personal”) keys. If you are using one institutional key for multiple Mac computers, go to Step 4.

4. Assign an Active Directory user who is authorized to manage an encrypted disk. FileVault 2 requires that you specify one or more “Managed By” users who can manage the encrypted disk, including the ability to lock and unlock it.

5. Enable the Enable FileVault 2 group policy. Enabling the “FileVault 2” group policy applies the FileVaultMaster certificate to Mac computers.

6. Set up and verify FileVault 2 protection. After FileVault 2 protection is enabled, the disk encryption process begins after the FileVault-authorized user logs off the computer.

Before You Begin Configuring Filevault 2

Be aware of the following requirements and limitations when configuring FileVault 2 through Delinea group policy:

- The Mac computer must be running OS X 10.9 or above.
- The Mac computer must have a recovery partition — generally, this partition is created by default during Mac OS X or macOS installation.
- FileVault 2 must *not* be enabled on the Mac computer (through the Security & Privacy System Preference).

If it is already configured, configuring FileVault 2 through Delinea Management Services for Mac will have no effect.

- Enabling FileVault 2 protection disables auto log on for the Mac computer.
- FileVault 2 protection does not support smart card authentication at start up of the computer.

  The Apple technical white paper, "Best Practices for Deploying FileVault2" provides more information about using FileVault 2; specifically, the section "Two Factor Authentication" discusses the limitations of using FileVault 2 with alternate authentication methods such as smart cards.

Create Filevault Master Keychain

The procedure described in this section is required only if you are using one institutional key for multiple Mac computers. If you are using computer-specific ("personal") keys, go to the section below, **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**.

On the Mac computer, you create a FileVault master keychain, which contains a private key that can be used to unlock the encrypted drive on the computer.

You can create the master keychain through the Mac user interface, or by executing commands in the Terminal application. Instructions are provided for each procedure.

> **Note**: If the computer already has a FileVault master keychain, you can skip this procedure and go to Export certificate from FileVault master keychain and upload it to a domain server.

To create a master keychain through the user interface

1. On a computer running OS X 10.9 or above, log on with an administrator's account and open **System Preferences**, then double-click **Users & Groups**.
2. If necessary, click the lock icon and enter credentials to authenticate.
3. Select an administrator's account, then click the service icon (⚙) and select **Set Master Password** from the pop-up menu.

4. Create a master password by typing it in **Master password** and re-typing in **Verify**.
5. Click **OK** to save the master password.

Setting a master password creates a keychain file in the following location:

/Library/Keychains/FileVaultMaster.keychain

This file contains the private key required to unlock the encrypted disc and is the only recovery method you will have for encrypted disc recovery. Store FileVaultMaster.keychain in a safe location, such as an external drive or an encrypted disk image on another physical disk.

To create a master keychain by executing commands in the Terminal application

1. On a Mac computer, open the Terminal application.
2. Run the following command:

   sudo security create-filevaultmaster-keychain
3. Enter the password for the root account when prompted as follows:

   To proceed, enter your password or type Ctrl-C to abort
4. Enter the master password to create when prompted to do so:

   password for new keychain
5. Retype the new master password when prompted to do so:

   retype password for new keychain

   You will see a message that the new password is being created:

   Generating a 2048 bit key pair; ...

Setting a master password creates a keychain file in the following location:

/Library/Keychains/FileVaultMaster.keychain

This file contains the private key required to unlock the encrypted disc and is the only recovery method you will have for encrypted disc recovery. Store FileVaultMaster.keychain in a safe location, such as an external drive or an encrypted disk image on another physical disk.

Export Certificate from Filevault Master Keychain and Upload it to a Domain Server

The procedure described in this section is required only if you are using one institutional key for multiple Mac computers. If you are using computer-specific ("personal") keys, go to the section below **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**.

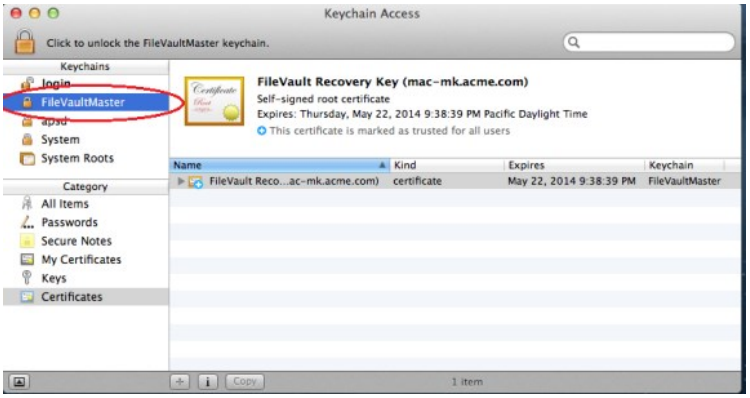
After you create a master password, as explained in the previous section, you must export the certificate associated with the master keychain to make it available for upload to the domain controller.

You can export the certificate by using the Mac user interface, or by executing commands in the Terminal application. Instructions are provided for each procedure.

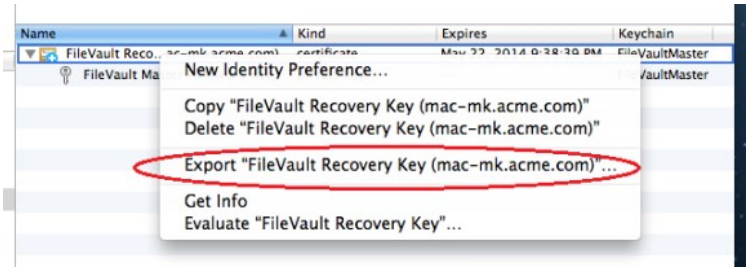
To export the certificate by using the Keychain Access utility

1. On the Mac computer, open the Keychain Access utility, or double-click the FileVaultMaster.keychain file, which is at the following location:

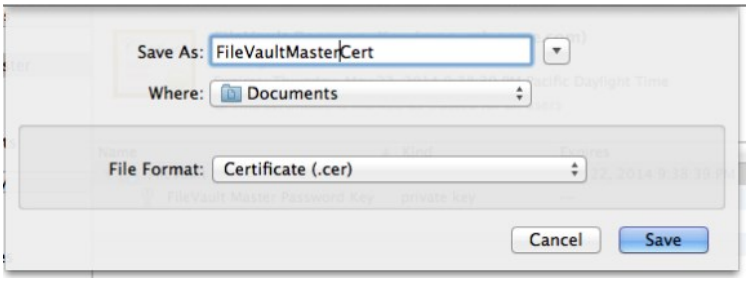
   /Library/Keychains/FileVaultMaster.keychain
2. Enter you password if prompted to do so.
3. In **Keychains**, select **FileVaultMaster**.



4. Select the certificate, **FileVault Recovery Key** in the right pane and expand it; then right-click and select **Export "FileVault Recovery Key"**.



5. Enter the following information for saving the certificate:



   - **Save As**: Type a name for the certificate, such as "FileVaultMasterCert".
   - **Where**: Navigate to a folder in which to save the certificate.
   - **File Format**: Select **Certificate (.cer)** from the scroll-down list.

     The certificate is now available for upload to a domain controller.

6. Copy the certificate to a location on a server that is accessible from the computer that you use to configure Group Policy for the domain.

   Later, when you enable the group policy to turn on FileVault 2 protection (see **Enable the Enable FileVault 2 Group Policy** below), you must be able to access this certificate from the domain controller on which you are running the Group Policy Editor.

To export the certificate by using Terminal commands

1. On the Mac computer, open the Terminal utility application.
2. Run the following command:

   ```
   sudo security export -k /PathToKeychain -t certs -f x509 -o /PathToCert
   ```

   Note: The sudo command is required only if FileVaultMaster.keychain is owned by root.

where:

- *PathToKeychain* is the path to FileVaultMaster.keychain; for example:

  /Library/Keychains/FileVaultMaster.keychain

- *PathToCert* is the path to the location in which to export the certificate; for example:

  /Documents/FileVaultMaster.cer

The certificate is now available for upload to a domain controller.

1. Copy the certificate to a location on a server that is accessible from the computer that you are using to configure Group Policy for the domain.

   Later, when you enable the group policy to turn on FileVault 2 protection (see **Enable the Enable FileVault 2 Group Policy** below), you must be able to access this certificate from the domain controller on which you are running the Group Policy Editor.

Enable Bitlocker Recovery Password Viewer in Active Directory

The procedure described in this section is required only if you are using computer-specific ("personal") keys. If you are using one institutional key for multiple Mac computers, go to **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**, below.

To enable the BitLocker Recovery Password Viewer feature in Active Directory

1. On the domain controller, open **Administrative Tools > Server Manager**.
2. In the navigation pane, right-click **Features** and select **Add Features**.
3. In the Add Features wizard, expand **Remote Server Administration Tools > Feature Administration Tools**, select **BitLocker Drive Encryption Administration Utilities**, click **Next**, and click **Install**.
4. After the BitLocker Drive Encryption Administration Utilities are installed, click **Close**.
5. To verify that the BitLocker Drive Encryption Administration Utilities are installed:
   1. Open Active Directory Users and Computers.
   2. Navigate to **domaincontroller > Domain Controllers**.
   3. In the right-hand ADUC pane, right-click the domain controller and select **Properties**.
   4. If the BitLocker Drive Encryption Administration Utilities installed correctly, the Properties dialog contains a **Bitlocker Recovery** tab. On that tab, a "No items in this view" message displays. That message is normal, and does not indicate a problem with the BitLocker Drive Encryption Administration Utilities installation.

Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk

Before enabling FileVault 2, you must assign a user account that is able to open the disk for the Mac computer after it is encrypted by FileVault 2. This setting specifies the "Managed By" user for a computer.

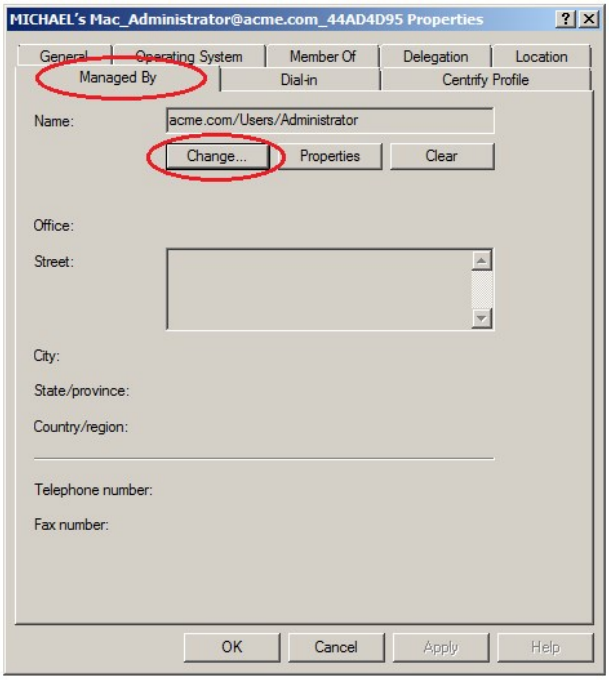
> **Note**: Enabling the "FileVault2" group policy, as explained in the next section, encrypts the entire disk for the computer. The user account that you assign in the current procedure will be authorized to access the disk during boot up so that this account will be able to log on. You can later add other accounts, but for now, this is the only account that will be able to log on to this computer.

The "Managed By" user account must be an Active Directory mobile user account. Configuring a Portable Home Directory for information about the steps you must take to create a mobile user account.

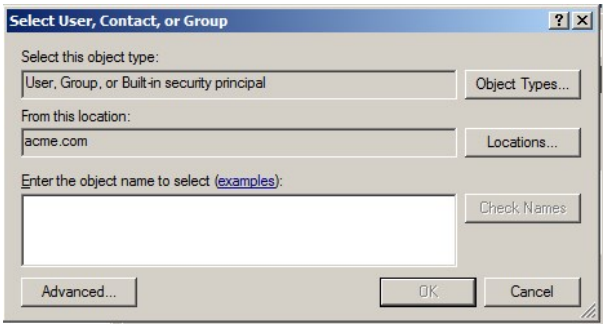
> **Note**: After you enable a user account to open an encrypted disk at start up, you cannot remove that account from the list. If you no longer want this user account to be able to unlock the disk, you can delete the account from Active Directory. Before doing so, be certain that you have at least one other account that can unlock the hard disk on this computer, otherwise you will no longer be able to access this computer.

To assign an account that can unlock the encrypted disk

1. On a domain controller, open Active Directory Users and Computers
2. Expand the domain object and navigate to the container that contains the Mac computer, for example, **Computers**.
3. Select the Mac computer that you plan to encrypt, right-click and select **Properties**.
4. Click the **Managed By** tab.



5. Click **Change**.
6. Enter the all or part of the name to search for (make certain that **User** is selected in **Object Type**) and click Check Names.



7. If the name is correct, click **OK** then **OK** again to save your changes.

Enable the Enable Filevault 2 Group Policy

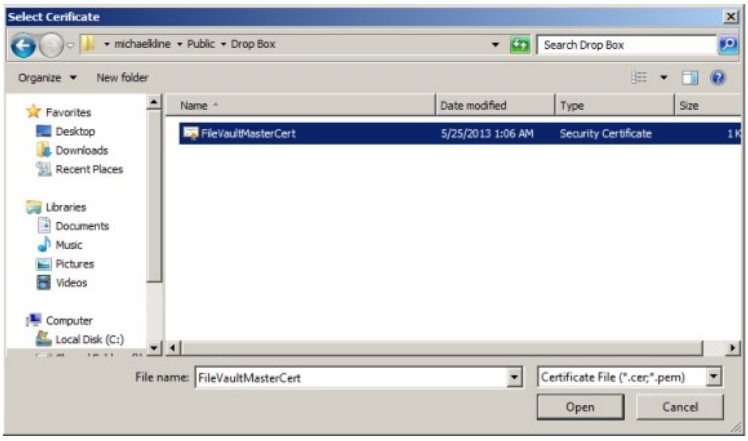
Next, enable the "Enable FileVault 2" group policy to encrypt the disk. When you enable this group policy, you select whether to use one institutional key for multiple Mac computers, or computer-specific ("personal") keys.

To enable the Enable FileVault 2 group policy

1. On a Windows computer, open the Group Policy Management Editor.
2. Select a Group Policy Object that applies to the Mac computer you are planning to encrypt, then right-click and select **Edit**.
3. Open **Computer Configuration > User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy**, then double-click **Enable FileVault 2**.
4. Click **Enable**.
5. Specify whether to use one institutional key for multiple Mac computers, or computer-specific ("personal") keys:

- To use one institutional key for multiple Mac computers, select **Use Institutional Recovery Key**. Then click **Select** to select the FileVault keychain certificate that you created earlier as described above in **Create FileVault Master Keychain**. If you select this option, the FileVaultMaster certificate is distributed to all of the Mac computers to which the group policy applies. Go to Step 6 and continue from there.
- To use computer-specific ("personal") keys, leave **Use Institutional Recovery Key** unchecked. In this situation, a personal recovery key is created for the Mac computer and stored in the computer object in Active Directory. The key is created and sent to the computer object in Active Directory after the "Managed By" user reboots the Mac computer (or restarts the agent), logs in, logs out, and provides the user password as described below in **Set Up and Verify FileVault 2 Protection**. The personal recovery key is used to enable FileVault2 protection on the Mac computer. Go to Step 8 and continue from there.

6. In the Explorer dialogue, navigate to the folder in which you uploaded the certificate.
7. Select the certificate and click **Open**.

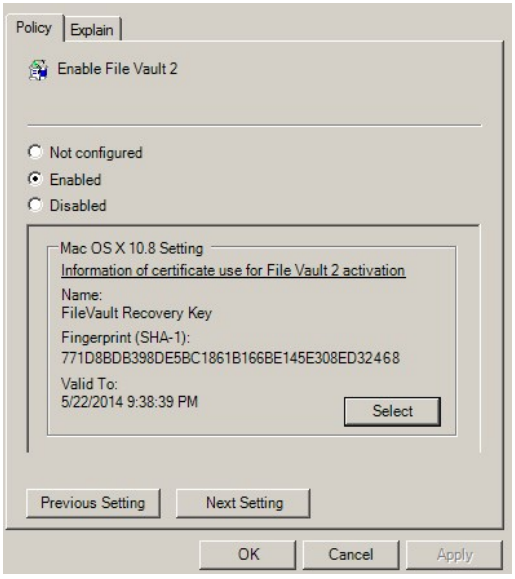


8. Click **OK** to enable the group policy.

   This group policy will automatically take effect at the next group policy update interval. To have it take effect immediately, run the following command in the Terminal application on the Mac computer:

   ```
   adgpudate
   ```

   If you selected **Use Institutional Recovery Key** in Step 5, the FileVaultMaster certificate name, a thumbnail, and the expiration date are displayed in the Group Policy.



   Note: The expiration date is not important because OS X does no revocation checking on this certificate.

The selected certificate should have the following usages: “Digital Signature”, “Key Encipherment”, “Data Encipherment” and “Key Certificate Sign”. If the certificate does not have these usages, an error message will appear:

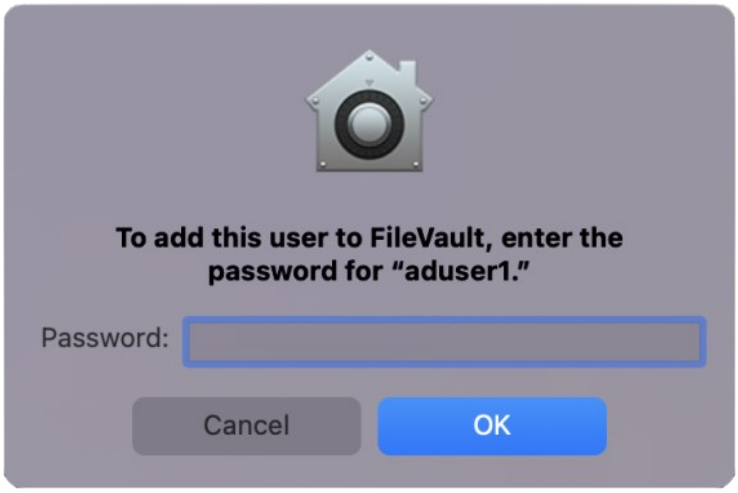
Set Up and Verify Filevault 2 Protection

FileVault 2 protects a Mac computer by encrypting the entire hard drive when a FileVault-authorized user (the “Managed By” user) logs out. To set up FileVault 2 for the first time, you must log on to the Mac computer as the “Managed By” user, then log out, as explained in the following procedure. After FileVault 2 is set up, only a FileVault 2-authorized user may start up the Mac computer. You may add more authorized users if you wish, or maintain a single account.

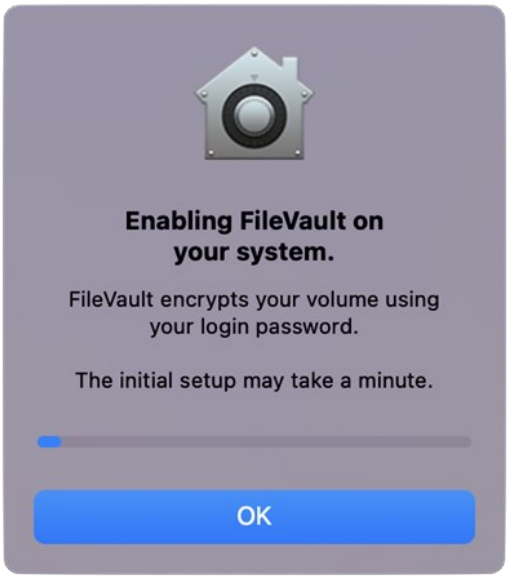
> **Note**: Although starting up the Mac computer requires a user account that is authorized to decrypt the start up disk, after the computer has started, this user account may log out to allow other user accounts to log in.

To set up FileVault 2 protection

1. Log on to the Mac computer with the “Managed By” account that you specified above in **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**.
2. Log the “Managed By” user out of the Mac computer, and when prompted, enter the user’s password to set up FileVault 2 protection.



The system displays a message that it is enabling FileVault protection, and when finished, restarts the computer.



3. Log back on to the Mac computer with the "Managed By" account.

   The log on screen will show the FileVault 2-authorized user alone, because this is the only user authorized to open the start up disk.

4. Open **System Preferences**, click **Security & Privacy** and click the **FileVault** tab to verify details about FileVault protection.

5. Log out the FileVault-authorized user.

   The log on screen now shows all users who are authorized for the computer.

A FileVault-authorized user is always required to start up the computer because the start up disk is encrypted. However, after the computer is running, any authorized user can log on to the computer. At this point, you have specified a single authorized account. To add more FileVault-authorized users, see the next section, **Adding FileVault-Authorized Users**.

Adding Filevault-Authorized Users

You can assign only one user as the "Managed By" user for the computer in Active Directory. If you want to authorize additional users to manage FileVault 2 protection, you must do so on the Mac computer by performing either one of the following procedures.

To authorize FileVault 2 users by using System Preferences

1. On the Mac computer, open **System Preferences > Security & Privacy**.
2. Click the **FileVault** tab, and if necessary, unlock the padlock.
3. Click the **Enable Users** button and an account list pops up.
4. Click **Enable Users** to add and enter password of that user.

To authorize FileVault 2 users by using Terminal commands

1. On the Mac computer, open the Terminal application.

2. Run the following command:

   ```
   sudo fdesetup add -usertoadd user1
   ```

   If prompted, enter the sudo password.

3. When prompted, enter the primary FileVault-authorized user name — this is the user who you specified to manage FileVault 2 (in the section above, **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**).

4. When prompted, enter the password for the primary FileVault-authorized user.

5. When prompted, enter the password for the new user who you specified on the command line (user1 in this example).

Changing FileVault 2 Settings

After you enable FileVault 2, the settings that you are most likely to change at a later time are the "Managed By" user and the FileVaultMaster certificate.

To change the "Managed By" user on a Mac computer

1. Disable FileVault 2 manually on the Mac computer as described below in **Disabling FileVault 2 Protection**.
2. On the domain controller, change the "Managed By" user as described in the section above, **Assign an Active Directory User Who is Authorized to Manage an Encrypted Disk**.
3. Ensure that the Mac computer can communicate with the domain controller (that is, it is in connected mode) so that it can fetch the new "Managed By" user information from Active Directory.

After you complete these steps, FileVault 2 protection is enabled on the Mac computer the next time the new "Managed By" user logs into the Mac computer.

To change the FileVaultMaster certificate

> **Note**: The procedure described in this section is supported only if you are using one institutional key for multiple Mac computers (that is, if you selected **Use Institutional Recovery Key** in **Enable the Enable FileVault 2 Group Policy**, above).

1. Disable FileVault 2 manually on each Mac computer that will use the new FileVaultMaster certificate. In most situations, this includes all computers to which the Enable FileVault 2 group policy is applied.
2. Specify a new FileVaultMaster certificate in the Enable FileVault 2 group policy as described in **Enable the Enable FileVault 2 Group Policy**, above.
3. Execute the adgpupdate command to have the Enable FileVault 2 group policy implement the new FileVaultMaster certificate on the Mac computers.

   If you do not execute adgpupdate, the old FileVaultMaster certificate is used until the next scheduled group policy update interval.

After you complete these steps:

- All Mac computers on which you disabled FileVault 2 (in Step 1) will use the new FileVaultMaster certificate the next time the "Managed By" user logs in.
- FileVault 2 protection is enabled on a Mac computer the next time the "Managed By" user logs into that Mac computer.

Disabling FileVault 2 Protection

The only way to disable FileVault 2 protection is manually on the Mac computer. You cannot disable it by disabling the Enable FileVault 2 group policy.

You can disable FileVault 2 protection through the Security & Privacy System Preference, or by issuing commands in the Terminal application — view one or the other of the two sets of instructions that follow.

To disable FileVault 2 protection by using Security & Privacy preferences

1. On the Mac computer, open **System Preferences > Security & Privacy** and click the FileVault tab.
2. Click the padlock and enter authentication information to unlock System Preferences.
3. Click **Turn Off FileVault**.
4. Click the padlock to secure the changes.
5. Restart the Mac computer.

The disk is no longer encrypted and all authorized users, not just FileVault-authorized users, should be visible on the log on screen.

To disable FileVault 2 protection by issuing Terminal commands

1. On the Mac computer, open the Terminal application.
2. Enter the following command:

   ```
   sudo fdesetup disable
   ```

3. Enter the root password when prompted.
4. Enter the password for the user account that is authorized to lock or unlock the disk.

   This is the password for the user who you assigned in Active Directory to manage the Mac OS X computer.

5. Restart the Mac computer.

The disk is no longer encrypted and all authorized users, not just FileVault-authorized users, should be visible on the log on screen.

What Happens if the Filevault-Authorized User's Password is Reset?

If the password is reset while the computer is off or not connected to the domain, the password will not be immediately updated so the user must first log in with the old password, then back in with the new password.

For example, follow these steps for a sample set up such as the following:

- The Mac computer is turned off.
- FileVault 2 is enabled.
- user1 is the primary FileVault 2 authorized user.

1. An administrator changes the user1 password in Active Directory Users and Computers (through Reset Password), and informs user1 of the change.
2. You start up the computer, log on as user1, and enter the new password, which fails.
3. Enter the old password, which works.
4. Restart the computer, log on and enter the new password, which should be successful.

Restoring the Filevault User List After Adflush

In Server Suite, if your FileVault 2 user list contains mobile users from another forest with one-way trust (that is, cross-forest mobile users), it is possible that those users will be removed from the FileVault 2 user list after you execute adflush or adflush -f.

After you upgrade to release 2015.1 or later, perform the following steps to ensure that cross-forest mobile users are added to the FileVault 2 user list permanently:

1. Execute the following command:

   ```
   adflush -f
   ```

   Executing this command removes the 2015-format, temporary GUID from cross-forest mobile users.

2. Execute the following command for each cross-forest mobile user that you want to add permanently to the FileVault 2 user list:

   ```
   adquery user -guid cross-forest-mobile-user-name
   ```

   Executing this command assigns a new, permanent GUID to each user that you specify.

3. Execute the following command for each cross-forest mobile user that you want to add to the FileVault 2 user list:

   ```
   fdesetup add -usertoadd cross-forest-mobile-user-name
   ```

   Executing this command adds the specified user to the FileVault 2 user list.

4. Execute the following command to verify that the users are added to the FileVault 2 user list:

   ```
   fdesetup list
   ```

How to Recover an Encrypted Disk

If a user forgets the password for their encrypted disk, you can unlock the disk for them using the institutional recovery key that you created. See the following two Web articles for information:

- Apple Support: "OS X: How to create and deploy a recovery key for FileVault 2".

  Note that you have already created the recovery key — you only need to read the information in the "Recovery" section.

- "Unlock or decrypt your FileVault 2-encrypted boot drive from the command line"

Deploy Configuration Profiles to Multiple Computers

This section explains how to deploy mobile configuration profiles to multiple computers by using a group policy setting (Install mobileconfig Profiles).

> **Note**: You can create mobile configuration profiles in a number of ways, for example by using the iPhone Config utility or OS X Server Profile Manager. This document assumes that you have already created a profile that you want to deploy, but does not show you how to do so.

You can deploy either computer or user profiles. For computer profiles, this feature requires OS X 10.7 or higher. For user profiles, this feature requires OS X 10.9 and higher.

The process for deploying a mobile configuration profile is as follows:

1. Create the mobile configuration profile.
2. Create a subdirectory in SYSVOL on the domain controller and copy the mobile configuration profile file to this directory. SYSVOL is a well-known shared directory on the domain controller that stores server copies of public files that must be shared throughout the domain.
3. Enable the "Install mobileconfig Profiles" group policy and specify the name of the file that you copied to SYSVOL.
4. The mapper script for the group policy runs on each Mac computer controlled by the GPO (when a user logs in or runs adupdate), downloads the profiles from the Active Directory server, and installs them in the Profiles system preference.

To create a subdirectory in SYSVOL:

1. Log in to the domain controller.
2. Change to the SYSVOL directory.

   For example, go to this directory:

   C:WindowsSYSVOLdomain

3. Create a new folder named mobileconfig.

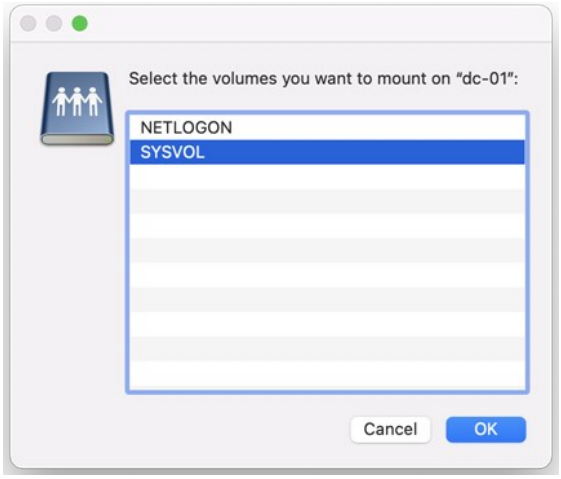
   > **Note**: Be certain that the name of the folder is exactly as shown in the step above. The group policy setting allows you to specify the name of the file but it always looks in SYSVOL\mobileconfig. Likewise, do not create sub-folders — the group policy does not look in sub-folders.

To copy configuration files to SYSVOL on the domain controller:

1. In the Finder on the Mac computer navigate to the folder that contains the profile to copy.
2. Select the file, for example, settings_for_all.mobileconfig and copy it to the desktop. When prompted, enter your administrator password to copy the file.
3. On the desktop, change the file permissions for settings_for_all.mobileconfig as follows, so you can copy it to SYSVOL:
   1. Select the file and click **File > Get Info**.
   2. In the dialog box, expand **Sharing & Permissions**, then click the lock icon and provide administrator credentials for making changes. Set the permissions for **everyone** to **Read only**.
   3. Reset the lock and close the open dialog.
4. On the Mac computer, copy the file from the desktop to SYSVOL on the Windows domain controller. If you are connected to the domain, you should see the domain controller in the Finder. If the domain controller is not visible in the Finder, connect to it:

   1. Click **Go > Connect to Server** and select the domain controller.

   2. When prompted select **SYSVOL**; for example:



3. Navigate to the mobileconfig directory you created, for example by clicking **acme.com** then **mobileconfig**.
4. Drag the settings_for_all.mobileconfig file to mobileconfig.

To configure the "Install MobileConfig Profiles" group policy:

1. On the Windows domain controller, open the Group Policy Management Editor and select the GPO that is used to manage Mac computers.
2. Navigate to **Computer Configuration > Policies > Mac OS X Settings > Custom Settings** and double-click **Install MobileConfig Profiles** to install a machine profile.

   To install a user profile, navigate to **User Configuration > Policies > Mac OS X Settings > Custom Settings** and double-click **Install MobileConfig Profiles**.
3. Select **Enabled**.
4. Click **Add**, then enter the name of the file that you copied to SYSVOL, for example, settings_for_all.mobileconfig.

   Be certain to include the .mobileconfig suffix.
5. Click **OK** to add the settings_for_all.mobileconfig file.
6. Click **OK** to enable the policy.

   This group policy will copy the settings_for_all.mobileconfg file, and install the profile, on every computer to which the GPO applies and that is joined to the domain. Note that after the profile is installed, it is deleted from the Mac computer.
7. Run the adgpupdate command on each target Mac computer to trigger an update of group policies and execute the new Install MobileConfig Profiles policy settings.

   By default, group policies are updated automatically every 90 minutes, so you can skip this step and wait for the automatic update if you wish.

Note the following about this process:

- If you add a profile file to SYSVOL, but do not specify it in the group policy setting, the profile will not be installed. Likewise, if you specify a file in the group policy that does not exist in SYSVOL, the profile will not be installed.
- If you add new files to the existing list in the group policy, those profiles will be installed — existing profiles will not be touched.
- If you remove a file from the group policy list (after the profile for the file was installed), the profile for that file will be uninstalled from the managed Mac computers.
- If you modify a file, the corresponding profile will be reinstalled.
- If two or more profile files have the same payloadIdentifier attribute, only one of them will be installed.
- If you change the group policy to "Disabled" or "Not Configured", all existing profiles that were installed previously by the group policy will now be uninstalled from the managed Mac computers.

  Note: The "Install MobileConfig Profiles" group policy only supports macOS 10.15 and lower.

Centrify group policies allow administrators to extend the configuration management capabilities of Windows Group Policy Objects to managed Mac computers and to users who log on to Mac computers. This chapter provides an overview to using the Delinea Mac group policies that can be applied to Mac computers and users

For reference information about the Mac OS X-specific computer and user policies that you can set, see the following topics.

- Setting Computer-based Group Policies
- Setting User-based Group Policies

For additional information about creating and using group policies and Group Policy Objects, see your Windows or Active Directory documentation, such as https://technet.microsoft.com/en-us/windowsserver/bb310732.aspx.

For information about other Centrify group policies that are not specific to Mac computers and users, see the *Group Policy Guide*.

The following topics are covered:

Understanding Group Policies and System Preferences

In many organizations, administrators who have both Windows and Mac computers in their organization want to manage settings for their Windows and Macintosh computers and users using a standard set of tools. In a Windows environment, the standard method for managing computer and user configuration settings is through Group Policy Objects applied to the appropriate site, domain, or organizational unit (OU) for different sets of computer and user accounts.

Delinea provides this capability for Mac computers and users through a group policy extension. The Delinea administrative template for Mac OS X (centrify_mac_settings.xml or centrify_mac_settings.adm) provides group policies that can be applied from a Windows server to control Mac OS X settings and behavior. These group policies can be applied to Mac OS X computers and to users who log on to those computers

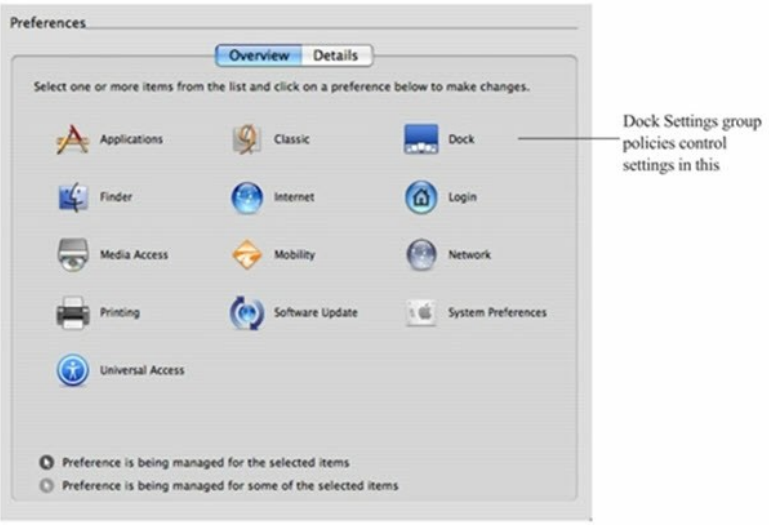
Through the Delinea administrative template for Mac OS X, Windows administrators using the Group Policy Management Editor can centrally access and control native Mac system preferences.

In the current Delinea administrative template for Mac OS X, Centrify group policies control settings for Personal, Hardware, Internet & Network, and System preferences, including:

- Accounts, (General) Appearance, Desktop & Screen Saver, Dock, Energy Saver, Network, Security & Privacy, Sharing, Software Update, and so on.

When you enable a group policy in a Windows Group Policy Object, you effectively set a corresponding system preference on the local Mac computer where the group policy is applied. For example, if you enable the group policy **Computer Configuration > Delinea Settings > Mac OS X Settings > Security > Require password to unlock each secure system preference**, it is the same as selecting the General tab of the Security & Privacy system preference, then clicking the **Require an administrator password to access system preferences with lock icons** option on a local Mac OS X computer. Once the group policy is enabled in the Windows Group Policy Object and updated on the local Mac computer, the corresponding option is checked:

In addition to the system preferences that are typically set on individual computers, there are many Mac configuration settings that are typically set from a Mac OS X server using the Workgroup Manager. These workgroup policies control application or media access, synchronization rules for mobile user accounts, the look and operation of the Dock, and other settings. The Delinea administrative template for Mac provides centralized access to many of these Workgroup Manager settings, including Applications, Dock, Media Access, Mobility, Software Update, and System Preferences.



Note: Not all group policies apply to all versions of the Mac operating environment or all computer models. If a particular system preference does not exist, is not applicable to the installed operating system, or is implemented differently on some computers, the group policy setting may be ignored or overridden by a local setting.

Group policies are available after you install the Delinea administrative template for Mac as described in **Installing the Administrative Template**, below. After you install the administrative template, the Windows administrator can use Active Directory MMC snap-ins or the Group Policy Management Console to create and link Group Policy Objects to sites, domains, or organizational units that include Mac computers that are joined to an Active Directory domain. Administrators can then use the Group Policy Management Editor to enable and configure the specific policies they want to enforce on Mac computers that are joined to the Active Directory domain.

See the *Group Policy Guide* for more information about using Active Directory Users and Computers or using the Group Policy Management Console or adding other Delinea administrative templates to a Group Policy Object.

Linking Group Policy Objects

To apply group policies to Mac computers, you can link an existing group policy object (GPO) that you are using for a Windows or UNIX computer, or create a new GPO to link to a domain or OU that contains your Mac computers and users. In general, it is recommended that you create an OU specifically for your Mac computers and link a new GPO to that OU. However, there is no problem adding the Mac group policies to an existing GPO and configuring policies for Mac computers; Mac OS X-specific policies that are applied to Windows or UNIX computers are simply ignored.

You apply GPOs to Mac users the same way; link the GPO to an OU containing the users. Group policies are only applied to users and computers in the organizational unit (OU) linked to the Group Policy object (GPO) and any of the child OUs. If your users and computers are in different OUs (which is common), Delinea recommends using user Group Policy loopback processing to make sure user policies are applied to everyone who logs on to a Mac. This is a standard Microsoft Group Policy that applies to every user to the computer. See Setting User-based Group Policies for more information about applying user policies.

Installing Mac Group Policies

Centrify group policies for Mac consist of two components

- The DirectControl agent for Mac and its associated configuration and system plug-in files that reside on the Mac computer. The DirectControl agent and related files determine the policies that have been applied to the local computer, or to the user who is logging on, and implement the policy through system preferences or other local configuration settings. This guide assumes that you have installed the DirectControl agent on your Mac computers.
- An administrative template (.xml or .admx file) that describes the policy settings available to the Group Policy Management Editor. The administrative template must be installed on a Windows computer that has the Group Policy Management Editor and the Delinea Group Policy Management Editor Extension. The Group Policy Management Editor and the Delinea Group Policy Management Editor Extension must be available for you to enable and configure policies. See the *Mac Quick Start Guide* for more information.

Installing the Administrative Template

Delinea provides templates in both XML and ADMX format. In most cases it is best to use the XML templates, which provide greater flexibility, such as the ability to edit settings after setting them initially, and in many cases contain validation scripts for the policies implemented in the template

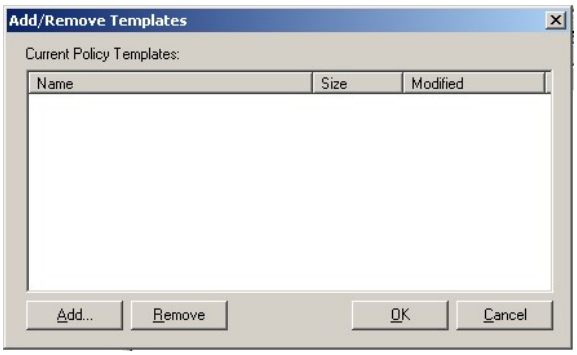
However, in certain cases, you may want to add templates by using the ADMX files. For example, if you have implemented a set of custom tools for the Windows ADMX-based policies, and want to extend those tools to work with the Delinea policies, you can implement the policies with ADMX template files. The Group Policy Management Editor will automatically read all ADMX files stored in the %systemroot%PolicyDefinitions folder.

The ADMX templates do not support extended ASCII code for locales that require double-byte characters. For these locales, you should use the XML templates.

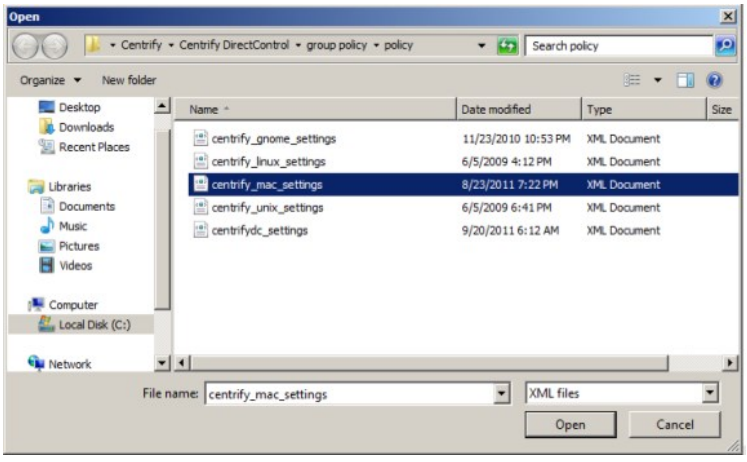
To install the Delinea XML administrative template for Mac group policies

This procedure assumes that you are using the Group Policy Management Console and have created a Mac OS X-specific GPO. For information about using a different console, such as ADUC, see the *Group Policy Guide*.

1. Open the Group Policy Management Console and select the Group Policy Object that you are using for Mac computers, right-click, then click **Edit** to open the Group Policy Management Editor.
2. Expand **Computer Configuration › Policies** and select **Delinea Settings**. Right click and click **Add/Remove Templates**.
3. Click **Add**, then navigate to the directory that contains the Delinea centrify_mac_settings.xml administrative template. By default, Delinea administrative templates are located in the C:Program Files\Common Files\Centrify Shared\Group Policy Management Editor Extension\policy folder.

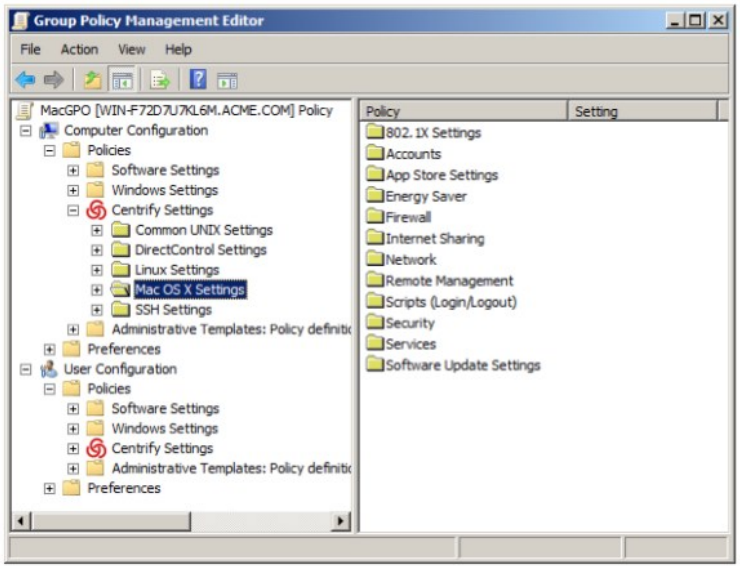


4. Select the centrify_mac_settings.xml file, then click **Open** to add this template to the list of Policy Templates.



5. Click **OK**.

You should now see the categories of Mac group policies listed as **Mac OS X Settings** under Delinea Settings in the Group Policy Management Editor. For example:



> **Note**: If you update Delinea to a new version, new templates may be included with the installation. To make any new policies included in the templates available for use, you must reapply each template by following the steps in one of these procedures. If you see the message, The selected XML file already exists. Do you want to overwrite it?, click **Yes**. This action overwrites the template with any new or modified group policies. It does not affect any configuration in the template that has been applied; that is, any policies that you have enabled remain enabled.

Setting Mac Group Policies

Like other group policies, policies for Mac users and computers are organized into categories within the Group Policy Management Editor under **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings** (Setting Computer-based Group Policies) or **Delinea Settings > Mac OS X Settings** (Setting User-based Group Policies). In general, these categories map directly to different types of Mac system preferences and individual policy settings within the categories map to specific settings within the system preference.

Normally, once enabled, policies get applied at the next group policy refresh interval, after the user logs out and logs back in, or after the computer has been rebooted. Some Mac group policies, however, require the user to log out and log back in or the computer to be rebooted. The description of each group policy indicates whether the policy can be applied "dynamically" at the next refresh interval or requires a re-login or a reboot.

You may also update group policies manually by running the adgpupdate command on an individual computer. See the next section, **Updating Configuration Policies Manually**.

> **Note**: The system preference updated on an individual computer must be closed, then reopened for the group policy setting to be visible.

In most cases, group policies can be Enabled to activate the policy or Disabled to deactivate a previously enabled policy. Changing a policy to Not Configured has no effect for any Mac group policies. Once a group policy is set on a local computer, it remains in effect even if the computer leaves the Active Directory domain. The administrator or users with an administrative account can change settings manually at the local computer, but any manual changes are overwritten when the group policy is applied.

Updating Configuration Policies Manually

Although there are Windows group policy settings that control whether group policies should be refreshed in the background at a set interval, Delinea also provides a command line program to manually refresh group policy settings at any time. This command line program, adgpupdate, forces the adclient daemon to contact Active Directory and collect group policy settings. With the adgpupdate command, you can specify whether you want to refresh computer configuration policies, user configuration policies, or both.

When you run the adgpupdate command, the adclient daemon does the following:

- Contacts Active Directory for computer configuration policies, user configuration policies, or both. By default, adclient collects both computer and user configuration policies.
- Determines all of the configuration settings that apply to the computer, the current user, or both, and retrieves those settings from the System Volume (SYSVOL).

- Writes all of the configuration settings to a virtual registry on the local computer.
- Starts the runmappers program to initiate the mapping of configuration settings using individual mapping programs for user and computer policies.
- Resets the clock for the next refresh interval.

For more information about using the adgpupdate command, see the adgpupdate page or *Using adgpupdate* in the *Administrator's Guide for Linux and UNIX*.

Applying Standard Windows Policies to Mac OS X

Every Group Policy Object includes several default Windows-based group policy categories and default Windows-based administrative templates for user and computer configuration. Most of the settings in the default Windows policies and administrative templates only apply to Windows computers and Windows user accounts. However, some of the common Windows configuration settings for password enforcement, such as the policies for minimum password length and complexity, do apply to Mac computers. If these settings are enabled for a Group Policy Object applied to a site, domain, or OU that includes Mac OS X computers, the settings are enforced for Mac users and computers.

The following sections describe the standard Windows group policies that you can apply to Mac computers and users and where you can find these policies when viewing a Group Policy Object in the Group Policy Management Editor.

Group Policy Refresh and Loopback Processing

The **Computer Configuration > Administrative Templates > System > Group Policy** object contains the following policies that you can use to control how group policies are refreshed and applied.

- Turn off background refresh of Group Policy
- Group Policy refresh interval for computers
- User Group Policy loopback processing mode

Synchronizing Time

By default, the local Network Time Protocol (NTP) Client is enabled and synchronizes your computer's clock to the Domain Controller. If you do not want your local NTP service to synchronize to the NTP service on the Domain Controller, explicitly disable the (Windows) Enable Windows NTP Client group policy. You can also synchronize to a different NTP server by specifying one in the Configure Windows NTP Client group policy.

To set these policies, in the Group Policy Editor, click **Computer Configuration > Administrative Templates > System > Windows Time Service > Time Providers**. The following policies are available to control time synchronization settings.

- Enable Windows NTP Client
- Configure Windows NTP Client

Specifying Time Sync Polling Interval

The **Computer Configuration > Administrative Templates > System > Windows Time Service > Global Configuration Settings** policy allows you to control the max polling interval with the MaxPollInterval option.

Configure Interactive Log On

Select the **Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options** object to configure the following policies related to interactive log on.

> **Note**: These policies apply to SSH login only, not to login through the graphical user interface.

- Interactive logon: Message text for users attempting to log on
- Interactive logon: Prompt user to change password before expiration

Set Password Requirements

Select the **Computer Configuration > Windows Settings > Security Settings > Account Policies > Password Policy** object to set password requirements.

- Enforce password history
- Maximum password age
- Minimum password age

- Minimum password length
- Password must meet complexity requirements
- Store passwords using reversible encryption

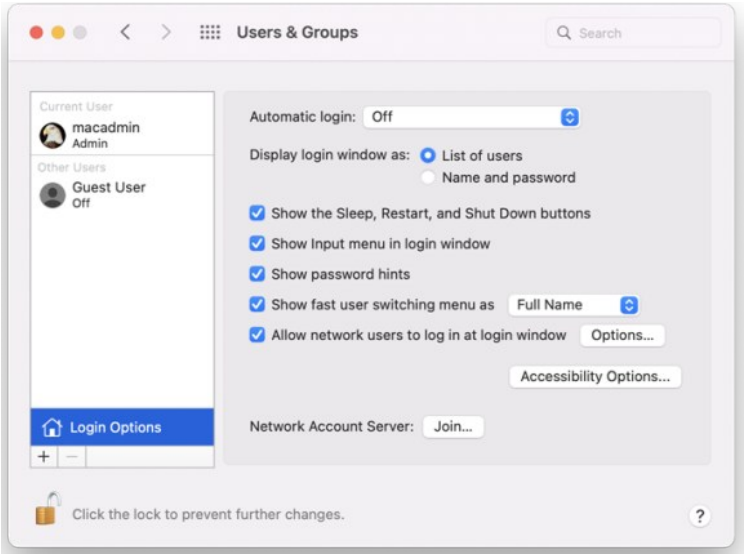
Configuring Mac-specific Parameters

Most configuration parameters apply to both Mac or only to actual UNIX or Linux systems. All these parameters are described in the *Configuration and Tuning Reference Guide*. However, the following parameters apply only to Mac OS X and are described in this section.

- adclient.autoedit.mac.netlogin
- adclient.mac.map.home.to.users
- adclient.network.wait.max
- mac.auto.generate.new.login.keychain
- mac.protected.keychain.enable
- mac.protected.keychain.user.default
- mac.protected.keychain.delete
- mac.protected.keychain.lock.inactivity
- mac.protected.keychain.lock.when.sleeping
- mac.keychain.sync.enabled
- mac.keychain.sync.polling.interval
- smartcard.pin.caching.disable
- logger.login.log

adclient.autoedit.mac.netlogin

System Preferences > Users & Groups (Accounts) has a login option: **Allow network users to log in at login window**:



If this option is deselected, Active Directory users will not be able to log into the computer. The configuration parameter adclient.autoedit.mac.netlogin controls whether this option can be deselected by users. By default, the parameter is true in the /etc/centrifydc/centrifydc.conf file:

```
adclient.autoedit.mac.netlogin: true
```

In this case, even if a user deselects the box, the box is selected again when adclient is restarted, effectively preventing a user from deactivating network login.

If you want to allow a user to deactivate network login, set the parameter to false. If a user deselects network login in **System Preferences > Accounts**, the next time adclient starts, network users will be unable to log in to the computer.

adclient.mac.map.home.to.users

On some versions of Mac OS X, /home is an automount point. If a zone user's home directory is set to /home/username, the operating system cannot create the

home directory and the user cannot log in. Therefore, you should not specify /home/username as the home directory for any Mac OS X users, but since this is a typical UNIX home directory, there may be Active Directory users who have a /home/username home directory.

To avoid potential problems, you can configure Delinea to change /home/username to /Users/username (the default Mac OS X home directory), in one of two ways:

- Enable the group policy, Map /home to /Users.
- Set the parameter, adclient.mac.map.home.to.users to true to enable the change for the local computer only; for example:

  adclient.mac.map.home.to.users:true

adclient.network.wait.max

The Delinea agent for Mac OS X performs network checks during startup to determine whether the device is connected to the domain. The adclient.network.wait.max parameter sets the maximum time the agent waits for the network before deciding to boot in either connected or disconnected mode. The default value is five seconds.

If DNS latency is high in your environment, the agent might determine that the device is in a Disconnected state too soon.

You can increase the value for the adclient.network.wait.max parameter if it's appropriate for your network environment; however, this might result in increased boot times.

logger.login.log

Login events are captured in /var/log/centrifydc-login.log by default. You can turn off this feature by setting the logger.login.log parameter to off. Refer to Collecting Information Specific to Login Events for more information about /var/log/centrifydc-login.

mac.auto.generate.new.login.keychain

Use this parameter to automatically generate a new login keychain if a user's keychain password does not match the password they used to successfully login.

The default value is false.

Refer to Auto Generate New Login Keychain for more information about the group policy that controls this parameter.

mac.protected.keychain.enable

Setting this parameter to true creates a new keychain protected by either an asymmetric token stored on a smart card or by a password, depending on the log in type.

The default value is false.

Refer to Enable Protected Keychain for more information about the group policy that controls this parameter.

> **Note**: Changing the group policy setting for this parameter does not change the parameter's value in this file. The two are set independently, with the group policy setting taking priority.

mac.protected.keychain.user.default

Setting this parameter to true sets the protected keychain as the default keychain for that user.

The default value is true.

Refer to Enable Protected Keychain for more information about the group policy setting that controls this parameter.

> **Note**: Changing the group policy setting for this parameter does not change the parameter's value in this file. The two are set independently, with the group policy setting taking priority.

mac.protected.keychain.delete

Setting this parameter to true deletes the existing password-protected Login Keychain after logging in.

The default value is false.

> **Note**: This parameter only works if mac.protected.keychain.enable is set to true.

Refer to Enable Protected Keychain for more information about the group policy setting that controls this parameter.

> **Note**: Changing the group policy setting for this parameter does not change the parameter's value in this file. The two are set independently, with the group policy setting taking priority.

mac.protected.keychain.lock.inactivity

Use this parameter to set the period of inactivity in minutes to automatically lock the protected keychain.

The default value is 0, which means the protected keychain is never automatically locked.

Refer to Lock Protected Keychain after Number of Minutes of Inactivity for more information about the group policy that controls this parameter.

> **Note**: Changing the group policy setting for this parameter does not change the parameter's value in this file. The two are set independently, with the group policy setting taking priority.

mac.protected.keychain.lock.when.sleeping

Setting this parameter to true locks the protected keychain when the Mac sleeps.

The default value is false.

Refer to Lock Protected Keychain When Sleeping for more information about the group policy that controls this parameter.

> **Note**: Changing the group policy setting for this parameter does not change the parameter's value in this file. The two are set independently, with the group policy setting taking priority.

mac.keychain.sync.enabled

This configuration parameter enables Keychain synchronization for the users on a mac.

If this parameter is enabled, the current login user will receive a password change notification when his/her password is changed remotely. When the user clicks on the notification, the Delinea Keychain Sync utility appears and allows the user to synchronize the Keychain password.

> **Note**: Password changes can only be detected when the machine is in connected mode.

The default value is false.

Refer to Enable Keychain Synchronization for more information about the related group policy.

mac.keychain.sync.polling.interval

This configuration parameter sets the password change detection interval when mac.keychain.sync.enabled is enabled.

This parameter determines the time (in minutes) between checking for changed passwords. There is a random zero to five minute variance in the actual interval each device is checked for a changed password to maintain performance. As a result, the minimum interval is five minutes.

> **Note**: Valid intervals are between 5 minutes and 1440 minutes (1 day).

The default value is 30.

Refer to Enable Keychain Synchronization for more information about the related group policy.

Centrify group policies allow administrators to extend the configuration management capabilities of Windows Group Policy Objects to managed Mac computers and to users who log on to Mac computers. This chapter provides reference information for the Centrify Mac group policies that can be applied specifically to Mac computers

The computer-based group policies are defined in the Centrify Mac administrative template (centrify_mac_settings.xml) and accessed from **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings**. See Understanding Group Policies for Mac Users and Computers for general information about how Delinea uses group policies to manage Mac settings and for information on how to install the group policy administrative templates.

For reference information about user-based policies, see Setting User-based Group Policies.

For information, see Applying Standard Windows Policies to Mac OS X and Configuring Mac-specific Parameters.

> **Note**: For more complete information about creating and using group policies and Group Policy Objects, see your Windows or Active Directory documentation. For more information about adding and using other Centrify group policies that are not specific to Mac computers and users, see the *Group Policy Guide*

Setting Computer-Based Policies for Mac

The following table provides a summary of the group policies you can set for Mac computers. These group policies are in the Centrify Mac administrative template (centrify_mac_settings.xml) and accessed from **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings**.

Allow Certificates with No Extended Key Usage Certificate Attribute	For smart card log in, allow the use of certificates that do not contain the extended key usage (EKU) attribute. This is a Windows policy that is defined in the **Administrative Templates > Windows Components > Smart Card** folder using an adm template.
Map /home to /Users	Map the /home/*username* directory to /Users/*username*. This is a Mac OS X-specific policy but defined in the **Direct Control Settings > Adclient Settings** folder using the centrifydc_settings.xml template.
802.1x Settings	Create login and system profiles for wireless authentication. These group policies correspond to 802.1X Options in the **Networks** system preference.
Accounts	Control the look and operation of the login window on Mac computers and map zone groups to the local administrator group. These group policies correspond to Login Options in the **Accounts** system preference.
App Store Settings Deprecated	Control the users and groups who can access the App Store. These group policies correspond to settings in the Sleep and Options panes in the **Energy Saver** system preference.
Custom Settings	Customize and install configuration profiles.
Energy Saver	Control sleep and wake-up option on Mac computers. These group policies correspond to settings in the Sleep and Options panes in the **Energy Saver** system preference.
Firewall	Control the firewall configuration on Mac computers. These group policies correspond to settings in the Firewall pane of the **Sharing** system preference.
Internet Sharing	Manage Internet connections on Mac computers. These group policies correspond to settings in the Internet pane of the **Sharing** system preference.
Network	Control DNS searching and proxy settings. These group policies correspond to settings in the TCP/IP and Proxies panes of the **Network** system preference.
Remote Management	Control Apple Remote Desktop access for zone users. These group policies correspond to the **Manage > Change Client Settings** options in Apple Remote Desktop.

Security & Privacy	Control security settings on Mac computers. These group policies correspond to settings in the **Security** system preferences.
Services	Control access to various services on Mac computers. These group policies correspond to settings in the Services pane of the **Sharing** system preference.
Software Update Settings	Control the options for automatic software updates on Mac computers. These group policies correspond to settings in the **Software Update** system preference.

For information about specific policies and how to set them, see the policy description (Explain text) or the corresponding discussion of the specific system preference or individual setting in the Mac Help.

Allow Certificates with no Extended Key Usage Certificate Attribute

Path

Computer Configuration › Policies › Administrative Templates: Policy Definitions› Windows Components› Smart Card.

Description

The group policy, "Allow certificates with no extended key usage certificate attribute" is defined in a Windows administrative template file (.adm), not in centrify_mac_settings.xml, and is in Administrative Templates, not in Mac Settings.

To enable or disable this policy, click **Computer Configuration › Policies › Administrative Templates: Policy Definitions › Windows Components › Smart Card**.

Enabling this policy setting allows the use of certificates for smart card login that do not have the Extended Key Usage (EKU) attribute set. Normally, certificates that are used for smart card login require this attribute with a smart card logon object identifier.

When you enable this policy, it sets the smartcard.allow.noeku parameter to true in the Centrify configuration file. Certificates with the following attributes can also be used to log on with a smart card:

- Certificates with no EKU
- Certificates with an All Purpose EKU
- Certificates with a Client Authentication EKU

If you disable or do not configure this policy setting (and do not set the smartcard.allow.noeku parameter to true in the Centrify configuration file) only certificates that contain the smart card logon object identifier can be used with smart card log in.

After changing the value of this parameter, you must re-enable smart card support by running the following sctool command as root:

```
[root]$ sctool -E
```

> **Note**: You must also specify the --altpkinit or --pkinit parameter when you run sctool with the -E option.

Map /home to /Users

Path

Computer Configuration › Policies › Centrify Settings › DirectControl Settings › Adclient Settings.

Description

The Mac group policy, Map /home to /Users is defined in the centrifydc_settings.xml file, not in centrify_mac_settings.xml, and is in Delinea Settings, not in Mac Settings.

To enable or disable this policy, click **Computer Configuration › Policies › Centrify Settings › DirectControl Settings › Adclient Settings**.

On some versions of Mac OS X, /home is an automount point. If a zone user's home directory is set to /home/*username*, the operating system cannot create the home directory and the user cannot log in. Therefore, you should not specify /home/*username* as the home directory for any Mac users, but since this is a typical UNIX home directory, there may be Active Directory users who have a /home/*username* home directory. To avoid potential problems, enable this

group policy, Map /home to /Users, to configure Delinea to change /home/*username* to /Users/*username* (the default Mac home directory). If you do not enable this policy, the change does not take effect.

This policy modifies the adclient.mac.map.home.to.users parameter in the Centrify configuration file.

802.1X Settings

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Settings** to create profiles for wireless network authentication. The profiles you specify with these group policies are created in the Network system preferences pane.

Enable Machine Ethernet Profile

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings > Enable Machine Ethernet Profile

Description

Enable this policy to create an 802.1X ethernet profile so users can authenticate to an 802.1X-protected network by using the specified machine certificate.

> **Note**: This group policy only supports macOS 10.15 and lower.

This policy supports the TLS protocol for certificate-based authentication for computers.

Before you can enable this policy, you must have a Windows server configured for 802.1X wireless authentication. The configuration includes certificate templates that are configured for auto-enrollment of domain computers and automatically downloaded to Mac computers when they join the domain. See Configuring 802.1X Wireless Authentication for details about what you must configure before enabling the current policy.

After enabling this policy, set the following:

- **Template Name**: Type the name of the auto-enrollment machine certificate that has been pushed down from the Windows domain server.

  When pushed to a Mac computer, certificate names are prepended with auto_; for example:

  auth_Centrify-1X

This group policy runs a script that looks for the specified certificate template in the /var/centrify/net/certs directory (which contains the certificate templates pushed down from the domain controller) and creates an Ethernet profile from this certificate.

Once enabled, this policy takes effect dynamically at the next group policy refresh interval.

Enable Machine Wi-Fi Profile

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings Enable Machine Wi-Fi Profile

Description

Enable this policy to create an 802.1X Wi-Fi profile for wireless network authentication for a computer.

> **Note**: This group policy only supports macOS 10.15 and lower.

This policy supports WEP or WPA/WPA2 security with the TLS protocol for certificate-based authentication for computers.

Before you can enable this policy, you must have a Windows server configured for 802.1X wireless authentication. The configuration includes certificate templates that are configured for auto-enrollment of domain computers and automatically downloaded to Mac computers when they join the domain. See Configuring 802.1X Wireless Authentication for details about what you must configure before enabling the current policy.

After enabling this policy, set the following:

- **SSID**: Type the SSID for the wireless network.
- **Template Name**: Type the name of the auto-enrollment machine certificate that has been pushed down from the Windows domain server.

  When pushed to a Mac computer, certificate names are prepended with auto_; for example: auth_Centrify-1X

  This group policy runs a script that looks for the specified certificate template in the /var/centrify/net/certs directory (which contains the certificate templates downloaded from the domain controller) and creates a WiFi profile from this certificate.
- **Security Type**: Select the security type from the drop-down list.
- **Other options**: Select one or more of the following options:
  - **Auto join**: Select this option to specify that the computer automatically join a Wi-Fi network that it recognizes. Do not select this option to specify that the logged in user must manually join a Wi-Fi network.
  - **Hidden network**: Select this option if the Wi-Fi network does not broadcast its SSID.
  - **Proxy PAC URL**: The URL of the PAC file that defines the proxy configuration. You can enter any string without spaces.
  - **Proxy PAC Fallback**: Allows the device to connect directly to the destination if the PAC file is unreachable. This option is disabled by default.

Once enabled, this policy takes effect dynamically at the next group policy refresh interval.

Enable User Ethernet Profile

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings > Enable User Ethernet Profile

Description

Enable this policy to create an 802.1X ethernet profile so users can authenticate to an 802.1X-protected network by using the specified user certificate.

> **Note**: This group policy only supports macOS 10.15 and lower.

This policy supports the TLS protocol for certificate-based authentication for users.

By default, the auto-enrolled user certificates are pushed down to ~/.centrify/autouser_(name).{cert.key.chain}. Certificates are also imported into each user's login keychain.

Before you can enable this policy, you must have a Windows server configured for 802.1X wireless authentication. The configuration includes certificate templates that are configured for auto-enrollment of domain computers and automatically downloaded to Mac computers when they join the domain. See Configuring 802.1X Wireless Authentication for details about what you must configure before enabling the current policy.

Users must perform these steps after login to authenticate to the network as the user:

1. Select **System Preferences > Network > Ethernet**.
2. If there are any pre-existing 802.1X connections, click **Disconnect** to disconnect the pre-existing connections. For example, if a machine 802.1X Ethernet policy has been set, the computer will already be authenticated using the machine credential.
3. Click Connect. This action prompts the user with a list of available user identities in *certificate-key* pair format.
4. Choose the appropriate auto-enrolled user identity.

Enable User Wi-Fi Profile

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings > Enable User Wi-Fi Profile

Description

Enable this policy to create an 802.1X Wi-Fi profile for wireless network authentication for a user.

> **Note**: This group policy only supports macOS 10.15 and lower.

This policy supports the TLS protocol for certificate-based authentication for users.

By default, the auto-enrolled user certificates are pushed down to ~/.centrify/autouser_(name).{cert.key.chain}. Certificates are also imported into each user's login keychain.

The resulting profile is signed using the first available auto-enrolled machine certificates, which are under /var/centrify/net/certs/auto_(name).{cert.key.chain}. If an auto-enrolled machine certificate is not available, the profile will be unsigned.

Before you can enable this policy, you must have a Windows server configured for 802.1X wireless authentication. The configuration includes certificate templates that are configured for auto-enrollment of domain computers and automatically downloaded to Mac computers when they join the domain. See Configuring 802.1X Wireless Authentication for details about what you must configure before enabling the current policy.

After enabling this policy, set the following:

- **SSID**: Type the SSID for the wireless network.
- **Security Type**: Select the security type from the drop-down list.
- **Other options**: Select one or more of the following options:
  - **Auto join**: Select this option to specify that the computer automatically join a Wi-Fi network that it recognizes. Do not select this option to specify that the logged in user must manually join a Wi-Fi network.
  - **Hidden network**: Select this option if the Wi-Fi network does not broadcast its SSID.
  - **Proxy PAC URL**: The URL of the PAC file that defines the proxy configuration. You can enter any string without spaces.
  - **Proxy PAC Fallback**: Allows the device to connect directly to the destination if the PAC file is unreachable. This option is disabled by default.

Users must perform these steps after login to authenticate to the network as the user:

1. Select **System Preferences > Network > Wi-Fi**.
2. If there are any pre-existing 802.1X connections, click **Disconnect** to disconnect the pre-existing connections. For example, if a machine 802.1X Ethernet policy has been set, the computer will already be authenticated using the machine credential.
3. Click **Connect**. This action prompts the user with a list of available user identities in *certificate-key* pair format.
4. Choose the appropriate auto-enrolled user identity (a *certificate-key* pair).

Specify System Profile (Deprecated)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings > Specify System Profile (Deprecated)

Description

This group policy is provided for backward compatibility with Mac OS X 10.6. If your environment does not contain any 10.6 computers, do not use this group policy.

Enable this policy to specify 802.1X system profile for wireless network authentication.

System profile can establish a wireless connection without a user login.

To add a system profile, enable the policy and click **Add** to enter the profile name and setting, then type a name for the profile.

The setting must follow this format:

- Network;Security Type;Authentication Method, where each field is separated by a semi-colon (;)
- Network is the wireless network name
- Security type is one of 802.1X WEP, WPA Enterprise, WPA2 Enterprise
- Authentication method is one or more of the following, separated by commas: TTLS, PEAP, LEAP, MD5

For example:

OFFICE1;WPA Enterprise;PEAP

OFFICE2;802.1X WEP;TTLS,PEAP

Automatically turn on Airport; to automatically turn on AirPort device if this type of profile is specified. Otherwise, the status of the AirPort device will not change.

Once enabled, this policy takes effect dynamically at the next group policy refresh interval.

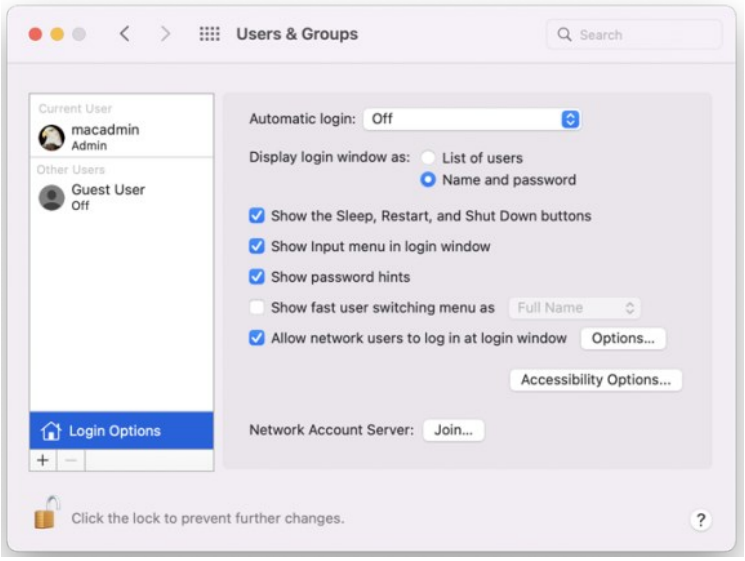
Accounts

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Accounts

Description

Use the **Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Accounts** settings to manage the options from the Accounts ( ) system preference on Mac computers. These group policies correspond to the options displayed when you select the **Accounts** system preference, then click **Login Options**. For example:



Set Login Window Settings

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Accounts › Set login window settings

Description

Configure the Login Options on a computer. If you enable this policy, you can configure the Login Window to:

- Display a text string as a login banner. The Banner you specify is displayed when the user is prompted to log on.
- Display a List of Users or a Name and Password field. Displaying the Name and Password requires users to provide their account name and password, and is more secure than displaying a list of user names.
- Show the Restart, Sleep, and Shut Down buttons.
- Show the Input menu in the login window to allow users to change the current Keyboard Layout.
- Show password hints in the login window.
- Use VoiceOver at the login window.
- Enable fast user switching.
- Display the HostName, IP Address, and OS X or macOS Version. Users need to click the clock in the top right corner to view each field.
- Disable reopening applications when logging back in. Check this to always uncheck the **Reopen Windows when logging back in** checkbox at

logout, restart, or shutdown.

- Hide all Local Users with a UID less than 500.
- Enabling the options in this group policy is the same as clicking **Login Options** in the Accounts system preference and setting the corresponding login window options.

  Note: This policy does not impact lock screens. It only impacts the login window.

  Note: If you click **Enable Fast User Switching**, this setting does not take effect until the Login Options in the Accounts system preference is opened manually by a user on the local host. This step is required to display the list of users in the upper-right corner of the menu bar. After users log on, the user's full name, short name, or icon identifier is displayed in the menu bar. If you want to change how users are displayed in the menu, you also must do so manually from the Login Options in the Accounts system preference.

Once enabled, this group policy takes effect when users log out and log back in or when the computer is rebooted.

Map Zone Groups to Local Admin Group

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Accounts > Map zone group to local admin group

Description

Specify one or more zone groups to map to the admin group on the local computer. Members of the groups you specify here have administrative privileges on the local computer, including:

- The use of sudo command in a shell
- The ability to unlock and make changes to System Preferences.

Be certain to create a zone group in Access Manager (or adedit) and add users who you want to have administrative privileges on managed Mac computers.

Note: If the local computer is connected to the domain through Auto Zone, you cannot create a zone group because there are no zones. However, all Active Directory groups are valid for the joined computer, so you can map any group to the local admin group, but you need to know the group's UNIX name, which you can retrieve on the local computer by using the adquery command, as shown in the following example.

```
[root]#adquery group -n
```

To set this policy

1. Open the policy and select **Enabled**.
2. Click **Add**.
3. Enter the name of a zone group in the box (or the UNIX group name if connected through Auto Zone). Then click **OK**.

Map Zone Groups to Local Group

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Accounts > Map zone group to local group

Description

Specify one or more zone groups to map to a Mac local group on the local computer. Members of the zone groups you specify here will be given the privileges of the local group on the local computer; for example:

- If you map to the _lpadmin and _lpoperator local groups, members of the zone group can manage printer settings on the local computer.
- If you map to the admin local group, members of the zone group obtain administrator privileges on the local computer.

  Note: To obtain administrator privileges for a zone group, you can either map to the local admin group with this policy, or use the Map zone groups to local admin group policy. However, do not do both as the results are unpredictable.

Be certain to create a zone group in Access Manager (or adedit) and add users who you want to have administrative privileges on managed Mac computers.

Note: If the local computers is connected to the domain through Auto Zone, you cannot create a zone group because there are no zones.

However, all Active Directory groups are valid for the joined computer, so you can map any group to the local admin group, but you need to know the group's UNIX name, which you can retrieve on the local computer, by using the adquery command, as follows

```
[root]#adquery group -n
```

To set this policy

1. Open the policy and select **Enabled**.
2. Click **Add**.
3. Enter the name of a local group and of a zone group in the respective boxes (or the UNIX group name if connected through Auto Zone), then click **OK**.

   You can repeat this step multiple times to map the zone group to more than one local group.

App Store Settings Deprecated

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > App Store Settings (Deprecated)

Description

Note: This policy has been deprecated and is no longer supported. Enabling it will have no effect. It is provided simply to allow you to disable the policy if it was set in an earlier version of the product. You can use the **Application Access Settings (deprecated)** group policies to control access to the App Store if you wish.

Prohibit Access to the App Store (Deprecated)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > App Store Settings (Deprecated) > Prohibit Access to the App Store (Deprecated)

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > App Store > Prohibit Access to App Store** group policy to control access to the App Store.

By default, all users can access the App Store. Enable this group policy to prohibit access to App Store to all users except the root user and those you specifically authorize with the options, **Allow these users to access App store**, and **Allow these groups to access App Store**.

You can set the following options with this policy:

Allow these users to access App Store	The names of local or AD users who are allowed to access the App Store. When this policy is enabled, only users on this list and the root user are allowed to access the App Store.
Allow these groups to access App Store	The names of local or AD groups that are allowed to access the App Store. When this policy is enabled, only users in the specified groups, and the root user, are allowed to access the App Store.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Custom Settings

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Custom Settings

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Custom Settings** group policy settings to customize

and install configuration profiles. The “Install MobileConfig Profiles” policy installs a device profile. To install a user profile, use the same policy in **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Custom Settings.**

Enable Profile Custom Settings

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Custom Settings > Enable profile custom settings

Description

Enable this group policy to use the Custom payload to specify preference settings for applications that use the standard plist format for their preference files.

> **Note**: This group policy only supports macOS 10.15 and lower.

You can use this GP to add specific keys and values to an existing preferences plist file. However, not all applications work with managed preferences, and in some cases only specific settings can be managed.

By default, you should place the plist files with preference settings in the folder \\domain\SYSVOL\<domain>\customsettings.

To add a file, click **Add** and enter name of a file that you placed in the SYSVOL location. The file you specify is relative to this path:

\\domain\SYSVOL\domain\customsettings

For example, if you enter:

com.apple.plist

the file that is imported is:

\\domain\SYSVOL\domain\customsettings\com.apple.plist

Install MobileConfig Profiles

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Custom Settings > Install MobileConfig Profiles

Description

Enable this group policy to install mobile configuration profiles on managed Mac computers.

> **Note**: There is a Computer Configuration version of this policy (which installs device profiles) and a User Configuration version (which installs user profiles).

> **Note**: This group policy only supports macOS 10.15 and lower.

Before enabling this policy, you must create a directory and copy mobile configuration files to SYSVOL on the domain controller. SYSVOL is a well-known shared directory on the domain computer that stores server copies of public files that must be shared throughout the domain.

Specifically, create the following directory on the domain controller:

\\domainName\SYSVOL\domainName\mobileconfig

and copy one or more mobile configuration profile files to this directory. See Deploy Configuration Profiles to Multiple Computers for details on how to do this.

To specify mobile configuration files to install, enable the policy, then click **Add**. Enter the name of a mobile configuration file that you placed in SYSVOL on the domain controller. Include the .mobileconfig suffix with the name.

If you specify a file that is not in the SYSVOL mobileconfig directory, the profile will not be installed.

If you add new files to the existing list in the group policy, those profiles will be installed — existing profiles will not be touched. If you remove previously specified files, the profiles defined by these files will be uninstalled.

If you add two or more profile files that have the same payloadIdentifier, only one of them will be installed.

If you change the group policy to “Disabled” or “Not Configured”, all existing profiles that were installed previously by the group policy will now be uninstalled

from the managed Mac computers.

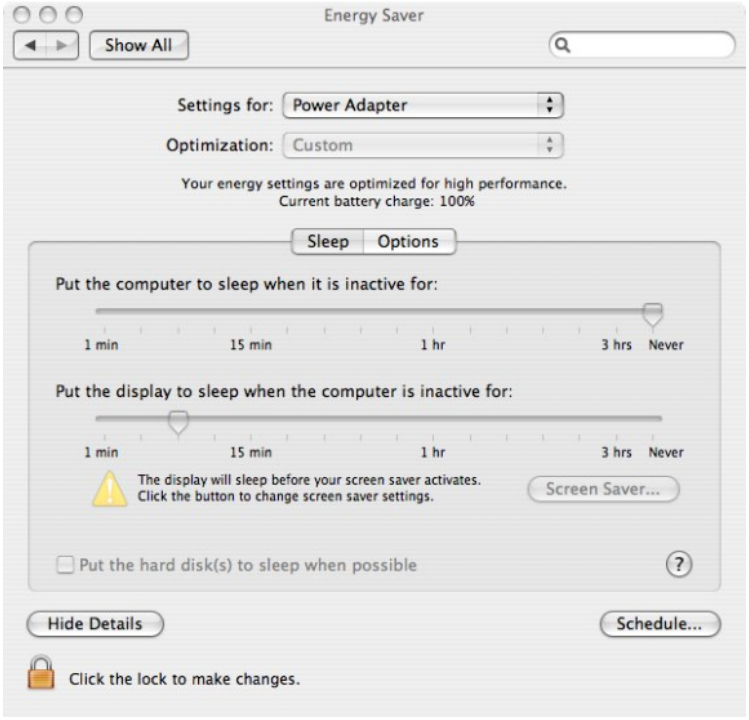
Energy Saver

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Energy Saver

Description

Use the **Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Energy Saver** settings to manage sleep and wake-up options from the Energy Saver ( ) system preference on Mac computers. For example:



You can configure power options or schedule startup and shutdown times.

Open the appropriate folder to set power options when running on AC power or battery power. Each folder has the identical set of group policies:

- On AC power
- On battery power

Allow Power Button to Sleep the Computer

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Energy Saver › Allow power button to sleep the computer

Description

Allow the power button to sleep the computer.

Enabling this group policy is the same as selecting the **Allow power button to sleep the computer** option in the Options pane of Energy Saver system preference.

This policy can take effect dynamically at the next group policy refresh interval.

Put The Hard Disk(s) to Sleep When Possible

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Put the hard disk(s) to sleep when possible

Description

Put computer hard disks to sleep when they are inactive.

Enabling this group policy is the same as selecting the **Put the hard disk(s) to sleep when possible** option in the Sleep pane of Energy Saver system preference.

This policy can take effect dynamically at the next group policy refresh interval.

Restart Automatically After a Power Failure

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Restart automatically after a power failure

Description

Enable to set the computer to automatically restart after a power failure.

Enabling this group policy is the same as selecting the **Restart automatically after a power failure** option in the Options pane of Energy Saver system preference.

This policy can take effect dynamically at the next group policy refresh interval.

Set Computer Sleep Time

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Set computer sleep time

Description

Specify the number of minutes of inactivity to allow before automatically putting a computer into the sleep mode.

If you enable this group policy, the period of inactivity you specify applies only when the computer is using its power adapter. If the computer is inactive for the number of minutes you specify, it is put in sleep mode.

Enabling this group policy is the same as selecting a time using the **Put the computer to sleep when it is inactive for** slider in the Sleep pane of Energy Saver system preference.

To prevent the computer from ever going into sleep mode, enter 0 for the number of minutes, or disable the policy.

This policy can take effect dynamically at the next group policy refresh interval.

Set Display Sleep Time

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Set display sleep time

Description

Specify the number of minutes of inactivity to allow before automatically putting the display into the sleep mode.

If you enable this group policy, the period of inactivity you specify applies when the computer is using its power adapter. If the computer is inactive for the number of minutes you specify, the display is put in sleep mode.

Enabling this group policy is the same as selecting a time using the **Put the display to sleep when it is inactive for** slider in the Sleep pane of Energy Saver system preference.

To prevent the display from ever going into sleep mode, enter 0 for the number of minutes, or disable the policy.

This policy can take effect dynamically at the next group policy refresh interval.

Wake When the Modem Detects a Ring

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Wake when the modem detects a ring

Description

Automatically take a computer out of sleep mode when the modem detects a ring. This group policy allows a computer that has been put to sleep to remain available to answer the modem.

This policy can take effect dynamically at the next group policy refresh interval.

Wake for Ethernet network administrator access

Automatically take a computer out of sleep mode when the computer receives a Wake-on-LAN packet from an administrator. This group policy allows a computer that has been put to sleep to remain available to network administrator access.

Enabling this group policy is the same as selecting the **Wake for Ethernet network administrator access** option in the Options pane of Energy Saver system preference.

This policy can take effect dynamically at the next group policy refresh interval.

Scheduled Events

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Scheduled events

Description

To configure sleep/shutdown times and startup times, open the Scheduled events folder (**Computer Configuration Policies > Centrify Settings > Mac OS X Settings > EnergySaver > Scheduled events**).

Set Machine Sleep/Shutdown Time

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Scheduled events > Set machine sleep/shutdown time

Description

Specify a time to shut down or put the computer to sleep.

Enabling this group policy is the same as selecting the **Schedule** button in the Energy Saver system preference, then specifying times and days to shut down or put the computer to sleep.

After enabling this policy, specify values for the following:

- **Action**: Select **sleep** or **shutdown**
- **Set machine sleep/shutdown time**: Enter a time in the format HH:mm using a 24 hour clock; for example, to shut down or put the computer to sleep at 10:05 P.M:

22:05

- **Sleep/shutdown machine on every:** Select the days of the week on which to shut down or sleep the computer at the specified time. All days are selected by default.

This policy can take effect dynamically at the next group policy refresh interval.

Set Machine Startup Time

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Energy Saver > Scheduled events > Set machine startup time

Description

Specify a time to start up the computer.

Enabling this group policy is the same as selecting the **Schedule** button in the Energy Saver system preference, then specifying times and days to start up the computer.

After enabling this policy, specify values for the following:

- **Set machine startup time**: Enter a time in the format HH:mm using a 24 hour clock; for example, to start up the computer at 7:55 A.M.:

07:55

- **Start machine on every**: Select the days of the week on which to start the computer at the specified time. All days are selected by default.

This policy can take effect dynamically at the next group policy refresh interval.

Firewall

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall** settings to manage the firewall options on Mac computers.

Enabling the Centrify firewall group policies is the same as setting options from **System Preferences > Security > Firewall**.

> **Note**: With the Centrify Firewall Group Policies, you can allow all incoming connections, or limit connections to the specified services and applications. You cannot block all connections:

In addition, group policies are available for the Advanced firewall settings, Enable Firewall Logging, and Enable Stealth Mode.

Enable Firewall

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable firewall

Description

Prevent incoming network communication to all services and ports other than those explicitly enabled for the services specified in the Services pane of the Sharing system preferences.

This group policy turns on default firewall protection.

- Block all incoming connections:

  Block all incoming connections except those required for basic Internet services, such as DHCP, Bonjour, and IPSec.

- Automatically allow signed software to receive incoming connections:

  Allows software signed by a valid certificate authority to provide services accessed from the network. This setting will not take effect if **Block all incoming connections** is selected.

This policy takes effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable iChat

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable iChat

Description

On Mac OS X Servers, enabling this policy has no effect. If the firewall is enabled, the iChat service is not allowed through the firewall.

If the firewall is enabled, enabling this group policy is the same as clicking the **On** checkbox to allow communication through the firewall for iChat Bonjour. If you do not enable this group policy, traffic for iChat Bonjour will be blocked from the local computer.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable iPhoto Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable iPhoto Sharing

Description

On Mac OS X Servers, enabling this policy has no effect. If the firewall is enabled, the iPhoto Sharing service is not allowed through the firewall.

If the firewall is enabled, enabling this group policy is the same as clicking the **On** checkbox to allow communication through the firewall for iPhoto Bonjour Sharing. If you do not enable this group policy, traffic for iPhoto Bonjour Sharing will be blocked from the local computer. Users will be able to access iPhoto collections on other computers, but the local computer cannot be used to serve any iPhoto collections.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable iTunes Music Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable iTunes Music Sharing

Description

Enabling this group policy is the same as clicking the **On** checkbox to allow communication through the firewall for iTunes Music Sharing.

On Mac OS X Servers, enabling this policy has no effect. If the firewall is enabled, the iTunes Music Sharing service is not allowed through the firewall.

If you do not enable this group policy, traffic for iTunes Music Sharing will be blocked from the local computer. Users will be able to access iTunes collections on other computers, but the local computer cannot be used to serve any iTunes collections.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Network Time

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable network time

Description

On Mac OS X Servers, enabling this policy has no effect. If the firewall is enabled, the Network Time service is not allowed through the firewall.

If the firewall is enabled, enabling this group policy is the same as clicking the **On** checkbox to allow communication through the firewall for Network Time. If you do not enable this group policy, traffic from the Network Time service will be blocked.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Block UDP Traffic

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Block UDP traffic

Description

Enabling this group policy is the same as clicking the **Block UDP Traffic** checkbox in the Advanced firewall settings.

This group policy does not block UDP communications that are related to requests initiated on the local computer.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Firewall Logging

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Firewall > Enable firewall logging

Description

Log information about firewall activity, including all of the sources, destinations, and access attempts that are blocked by the firewall. The activity is recorded in the secure.log file on the local computer.

Enabling this group policy is the same as clicking the **System Preferences > Security > Firewall** then clicking **Enable Firewall Logging** in the Advanced firewall settings.

On Mac OS X Servers, enabling this policy has no effect.

This policy takes effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Stealth Mode

Prevent uninvited traffic from receiving a response from the local computer.

Enabling this group policy is the same as clicking the **System Preferences > Security > Firewall** then clicking **Enable Stealth Mode** in the Advanced firewall settings.

If you enable this group policy, the local computer will not respond to any network requests, including ping requests. Because the computer will not reply to ping requests, using this policy may prevent you from using network diagnostic tools that require a response from the local computer.

On Mac OS X Servers, enabling this policy has no effect.

This policy takes effect dynamically at the next group policy refresh interval without rebooting the computer.

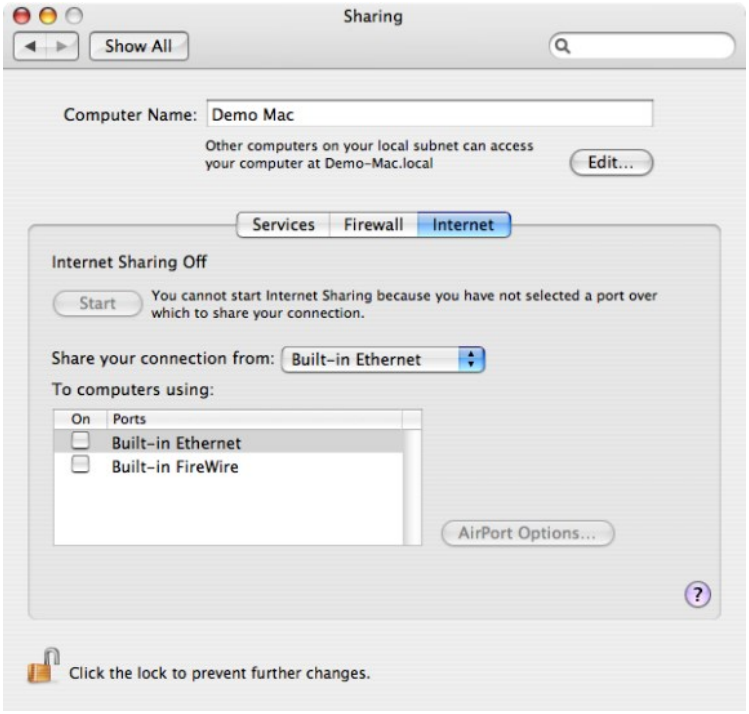
Internet Sharing

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Internet Sharing

Description

Use the **Computer Configuration › Policies › Centrify Settings › Mac OS X Settings ›Internet Sharing** group policy to prevent any kind of Internet sharing on the local computer. This group policy can only be used to prevent Internet sharing. Although this group policy corresponds to a setting on the Internet pane of the Sharing ( ) system preference, you can not use it to start Internet sharing, configure the shared connection, or set any other options. For example:



Disallow All Internet Sharing

Path

Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Internet Sharing › Disallow all Internet Sharing

Description

Prevent any kind of Internet sharing on the local computer. Enabling this group policy is the same as clicking **Stop** to prevent other computers from sharing an Internet connection on a local computer in the Internet pane of the Sharing system preference.

For this group policy, clicking Disabled or Not Configured has no effect. If you have previously Enabled the group policy, Internet sharing will remain off until you manually start it on the local computer.

Once enabled, this group policy takes effect when users log out and log back in, or dynamically at the next group policy refresh interval without rebooting the computer.

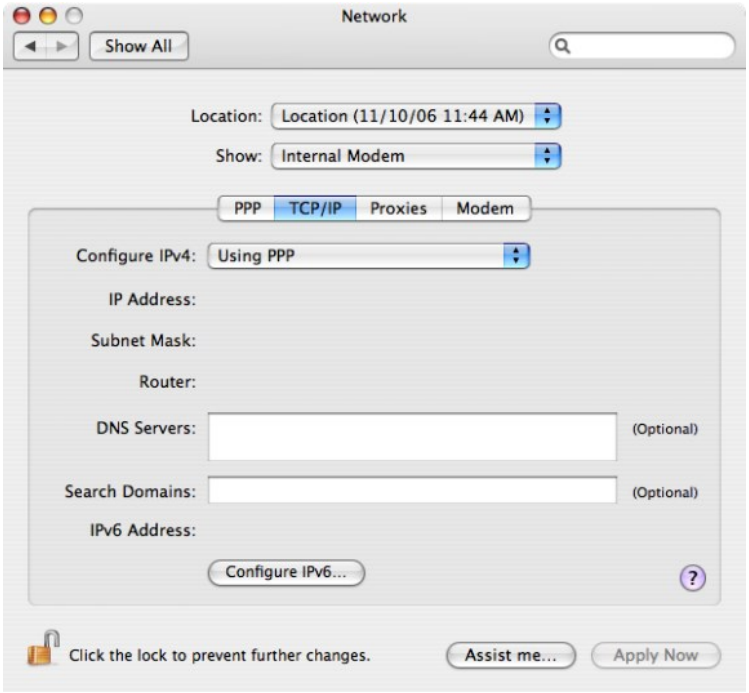
Network

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network** settings to manage DNS search requests and proxy settings. These group policies correspond to settings in the TCP/IP and Proxies panes of the Network ( ) system preference on Mac computers. For example:



Legacy Location Settings

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings

Description

Use **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings** to configure network settings for the Automatic network location.

Adjust List of DNS servers

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Adjust list of DNS servers

Description

Control the list of DNS servers when performing DNS lookups.

To use this policy, click **Enabled**, then click **Add**, type the IP address for a DNS server, then click **OK** to add the server to the list of DNS servers. Add as many servers as you want in this manner. When you are finished adding the servers, click **OK** to close the dialog box.

At any time while the policy is enabled, you can select an address in the list and click **Edit** to change the address, or **Remove** to remove it as a DNS server.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Adjust List of Searched Domains

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Adjust list of searched domains

Description

Control the list of domains to search when performing DNS lookups.

To use this policy, click **Enabled**, then click **Add**, type a domain name, then click **OK** to add the domain to the list of domains to search. Add as many domains as you want in this manner. When you are finished adding the domains to search, click **OK** to close the dialog box.

At any time while the policy is enabled, you can select a domain in the list and click **Edit** to change the name, or **Remove** to remove it as a domain to be searched.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Configure Proxies

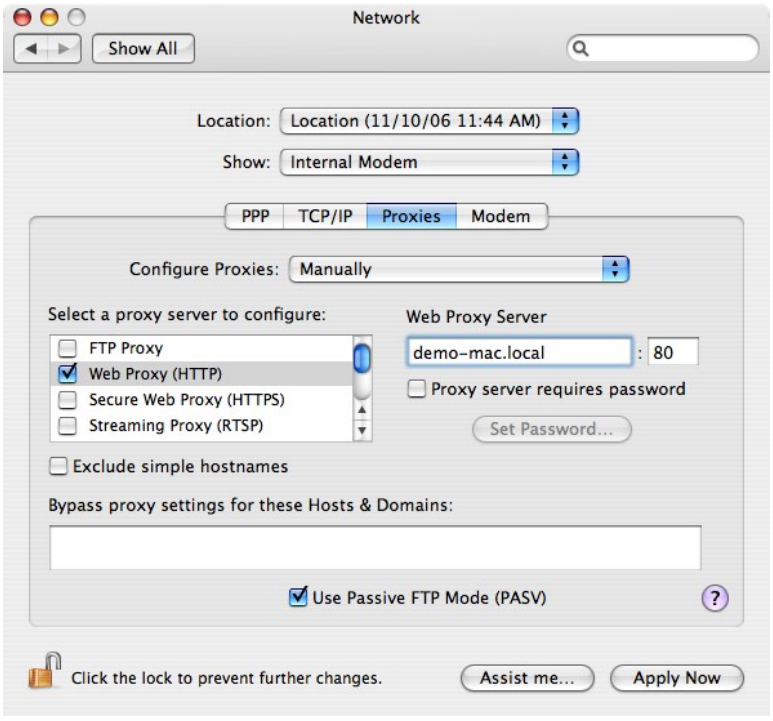
Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Configure Proxies

Description

Configure proxy servers to provide access to services through a firewall.

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Configure Proxies** settings to manage settings on the Proxies panes of the Network system preference. For example:



These group policies enable you to configure the host names (or IP addresses) and port numbers for the computers providing specific services, such as File Transfer Protocol (ftp), Hypertext Transfer Protocol (http), and HTTP over Secure Sockets Layer (https), through a firewall. A proxy server is a computer on a local network that acts as an intermediary between computer users and the Internet to ensure the security and administrative control of the network.

Enable Proxies

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Configure Proxies > Enable Proxies

Description

Configure the host name (or IP address) and port number for the computers providing specific services. Within this category, you can enable the following proxy servers:

- Use the **Enable FTP Proxy** policy to configure the host name and port number for the FTP proxy server (FTP protocol).
- Use the **Enable Web Proxy** policy to configure the host name and port number for the Web proxy server (HTTP protocol).
- Use the **Enable Secure Web Proxy** policy to configure the host name and port number for the Secure Web proxy server (HTTPS protocol).
- Use the **Enable Streaming Proxy** policy to configure the host name and port number for the Streaming proxy server (RTSP protocol).
- Use the **Enable SOCKS Proxy** policy to configure the host name and port number for the Streaming proxy server (SOCKS protocol).
- Use the **Enable Gopher Proxy** policy to configure the host name and port number for the Gopher proxy server.
- Use the **Enable Streaming Proxy** policy to configure the host name and port number for the Streaming proxy server (RTSP protocol).
- Use the **Enable Proxies using a PAC file** policy to configure proxy servers from a proxy configuration file.

These policies can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Exclude Simple Hostnames

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Configure Proxies > Exclude simple hostnames

Description

Prevent requests to unqualified host names from using proxy servers. If you enable this policy, users can enter unqualified host names to contact servers directly rather than through a proxy.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Use Passive FTP Mode (PASV)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Configure Proxies > Use Passive FTP Mode (PASV)

Description

Use the FTP passive mode (PASV) to access Internet sites when computers are protected by a firewall.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Bypass Proxy Settings for these Hosts & Domains

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy location settings > Configure Proxies > Bypass Proxy settings for these Hosts & Domains

Description

Specify fully-qualified host names and domains for which you want to bypass proxy settings.

You should use this policy to define the hosts or domains that should never be contacted by proxy.

To use this policy, click **Enabled**, then click **Add**, type a host or domain name, and click **OK** to add the entry to the Show Contents list.

Each host or domain should be listed as a separate line in the Hosts and Domains list. For each host or domain, click **Add**, type the host or domain name, and click **OK** to add the host or domain as a new entry in the list. When you are finished adding items to the list, click **OK** to close the policy dialog box.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Location 1 and Location 2

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 2

Description

Use **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1** to configure network settings for an additional network location. The group policies in Location 2 are identical, and allow you to configure network settings for another network location.

Adjust List of DNS Servers

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1/Location 2 > Adjust list of DNS servers

Description

Control the list of DNS servers when performing DNS lookups.

To use this policy, click **Enabled**, then click **Add**, type the IP address for a DNS server, then click **OK** to add the server to the list of DNS servers. Add as many servers as you want in this manner. When you are finished adding the servers, click **OK** to close the dialog box.

At any time while the policy is enabled, you can select an address in the list and click **Edit** to change the address, or **Remove** to remove it as a DNS server.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Adjust List of Searched Domains

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1/Location 2 > Adjust list of searched domains

Description

Control the list of domains to search when performing DNS lookups.

To use this policy, click **Enabled**, then click **Add**, type a domain name, then click **OK** to add the domain to the list of domains to search. Add as many domains as you want in this manner. When you are finished adding the domains to search, click **OK** to close the dialog box.

At any time while the policy is enabled, you can select a domain in the list and click **Edit** to change the name, or **Remove** to remove it as a domain to be searched.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Network Location

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1/Location 2 > Enable network location

Description

Enable all network location settings under the current location category and set its location name. This policy must be enabled to apply settings in this location category (for example, Location1).

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Configure Proxies

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Location 1/Location 2 > Configure Proxies

Description

Configure proxy servers to provide access to services through a firewall. The group policies in this folder are the same as the ones in Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Network > Legacy Location settings> Configure Proxies. Refer to Configure Proxies for more information.

Remote Management

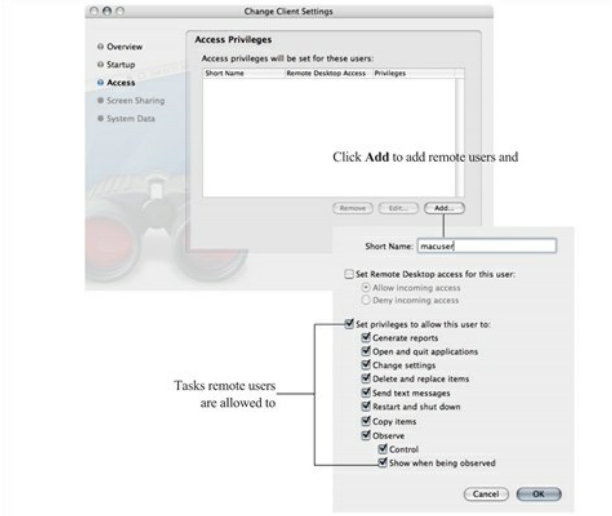
Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Remote Management

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Remote Management** settings to control Apple Remote Desktop access for zone users. You can use these group policies to give Active Directory group members permission to remotely control Mac computers without physically having to activate the Apple Remote Desktop on the remote Mac computer.

The Remote Management group policies correspond to the **Manage > Change Client Settings** options in Apple Remote Desktop and are similar to access privileges defined on a client computer using the Sharing system preference. For example:



> **Note**: Because the group policies correspond to the **Manage > Change Client Settings** options in Apple Remote Desktop, the group policy settings are not displayed in the local system preference on the Mac client. Although the tasks you can assign to different groups by group policy correspond to tasks you can assign using the local Sharing system preference on a Mac client computer, the group policy settings do not update the local system preference to display check marks for the tasks that the remote users have been given permission to perform.

Enable Administrator Access Groups

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Remote Management > Enable administrator access groups

Description

Allow all users who are members of the following Apple Remote Desktop administrator groups to access this computer using Apple Remote Desktop.

Before enabling this group policy, you should create each Active Directory security group you intend to use and add a UNIX profile for each group to the zone, using the exact UNIX group names (ard_admin, ard_reports, ard_manage, ard_interact).

> **Note**: Creating UNIX profiles with these group names displays a warning message because the names are longer than eight characters. You can safely ignore this warning message.

Enabling this policy allows users in the following groups to manage Mac computers through Apple Remote Desktop:

- ard_admin gives all members of the group the ability to remotely control the computer desktop.
- ard_reports gives all members of the group the ability to remotely generate reports on the computer.
- ard_manage gives all members of the group the ability to manage the computer using Apple Remote Desktop. Users in this group can perform the following tasks by using Apple Remote Desktop:
  - Generate reports
  - Open and quit applications
  - Change settings
  - Copy Items
  - Delete and replace items
  - Send text messages
  - Restart and shut down
- ard_interact gives all members of the group the ability to interactively observe or control the computer using Apple Remote Desktop.

  Users in this group can perform the following tasks by using Apple Remote Desktop:
  - Send text messages
  - Observe
  - Control

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

See Setting Up Local and Remote Administrative Privileges for information on how to use this group policy with the Map Zone Groups to Local Admin Group policy to enable both local and remote administrative access for the same group of users.

Scripts (Login/Logout)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout)

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout) > Specify multiple login scripts** group policy to deploy login scripts that run when an Active Directory or local user logs on. When you use this group policy, the login scripts are stored in the Active Directory domain's system volume (sysvol) and transferred to the Mac computer when the group policies are applied. Login scripts are useful for performing common tasks such as mounting and un-mounting shares

This policy is also available as a user policy. If you specify scripts using both the computer and user policies, the computer scripts are executed first.

Specify Multiple Login Scripts

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout) > Specify multiple login scripts

Description

Specify the names of one or more login scripts to execute when an AD or local user logs on. The scripts you specify run simultaneously in no particular order.

Before enabling this policy, you should create the scripts and copy them to the system volume (sysvol) on the domain controller. By default, the login scripts are stored in the system volume (SYSVOL) on the domain controller in the directory:

```
\\domain\SYSVOL\domain\Scripts
\scriptname1
\scriptname2
...
```

After enabling this policy, click **Add** and enter the following information:

- **Script**: The name of the script and an optional path, which are relative to \\domain\SYSVOL\domain\scripts\

  For example, if the domain name is ajax.org and you enter a script name of start.sh, the script that gets executed on the domain controller is:

  \\ajax.org\SYSVOL\ajax.org\Scripts\mlogin.sh

You can specify additional relative directories in the path, if needed; for example, if you type submlogin.sh, the file that gets executed is:
\\ajax.org\SYSVOL\ajax.org\Scripts\sub\mlogin.sh

- **Parameters**: An optional set of arguments to pass to the script. These arguments are interpreted the same way as in a UNIX shell; that is, space is a delimiter, and backslash is an escape character. You can also use $USER to represent the current user's name. For example:

```
arg1 arg2 arg3
arg1 'a r g 2' arg3
```

  Note: Be certain authenticated users have permission to read these files so the scripts can run when they log in.

Once this group policy is enabled, it takes effect when users log out and log back in.

Scripts (LaunchDaemons)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (launchDaemons)

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (LaunchDaemons) > Specify multiple LaunchDaemon scripts** group policy to deploy scripts that run when launchd starts (system boot up). When you use this group policy, the LaunchDaemon scripts are stored in the Active Directory domain's system volume (sysvol) and transferred to the Mac computer when the group policies are applied. Using LaunchDaemons to run scripts allows you to run the scripts as root, where the **Specify multiple login scripts** group policy can only be run as the logged in user.

Refer to the following Apple resources to learn more about Launch Daemons and Agents.

- https://developer.apple.com/library/content/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html
- https://developer.apple.com/library/content/technotes/tn2083/_index.html#//apple_ref/doc/uid/DTS10003794

Specify Multiple LaunchDaemon Scripts

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (launchDaemons) > Specify multiple LaunchDaemon scripts

Description

Enable this group policy to specify multiple scripts to run automatically when launchd starts (system boot up).

The scripts you specify run simultaneously in no particular order.

Before enabling this policy, you should create the scripts and copy them to the system volume (sysvol) on the domain controller. By default, the LaunchDaemon scripts are stored in the system volume (SYSVOL) on the domain controller in the directory:

```
\\domain\SYSVOL\domain\Scripts
\scriptname1
\scriptname2
...
```

After enabling this policy, click **Add** and enter the following information:

- **Script**: The name of the script and an optional path, which are relative to \\domain\SYSVOL\domain\scripts\.

For example, if the domain name is ajax.org and you enter a script name of startup.sh, the script that gets executed on the domain controller is:

\\ajax.org\SYSVOL\ajax.org\Scripts\startup.sh

You can specify additional relative directories in the path, if needed; for example, if you type submlogin.sh, the file that gets executed is:
\\ajax.org\SYSVOL\ajax.org\Scripts\sub\startup.sh

- **Parameters**: An optional set of arguments to pass to the script. These arguments are interpreted the same way as in a UNIX shell; that is, space is a delimiter, and backslash is an escape character. For example:

arg1 arg2 arg3
arg1 'a r g 2' arg3

Security & Privacy

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy

Description

Use the Centrify group policies found in **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy** to manage the Keychain, public and private keys, and the options from the Security & Privacy () system preference on Mac OS X computers.

Auto Generate New Login Keychain

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Auto Generate New Login Keychain

Description

Use this policy to automatically generate a new login keychain if a user's keychain password does not match the password they used to successfully login, resulting in the message "the system was unable to unlock your login keychain".

This commonly occurs if someone has changed their account password on another system.

If this policy is enabled, a new keychain will be generated when a password sync issue is discovered. This new keychain will be set as the default login keychain and the previous keychain will be moved to a backup.

Delinea recommends disabling this policy if you plan to use Enable Keychain Synchronization.

This policy is disabled by default.

Certificate Validation Method

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Certificate validation method

Description

Specify the certificate validation method to use for the Mac computer.

> **Note**: This group policy has no effect on the "Keychain Access > Preferences > Certificates" settings. Keychain Access > Preferences are per-user settings, which are not used by a Mac computer during login. This group policy changes Centrify SmartCardTool > Revocation settings, which represent the system settings used by a Mac computer during login.

This policy allows you to choose either one, or both of the two common methods for verifying the validity of a certificate:

- **Certificate Revocation List**: Use a certificate revocation list (CRL) from a revocation server.
- **Online Certificate Status Protocol**: Use an online certificate status protocol (OCSP) responder to validate certificates.

  If you select this option, you can specify a local responder to override the one provided in the certificates.

For each validation option, you can select one of the following settings:

- **Off**: No revocation checking is performed.
- **Best attempt**: The certificate passes unless the server returns an indication of a bad certificate.

  This setting is recommended for most environments.
- **Require if cert indicates**: If the URL to the revocation server is provided in the certificate, this setting requires a successful connection to a revocation server as well as no indication of a bad certificate.

  Specify this option only in a tightly controlled environment that guarantees the presence of a CRL server or OCSP responder. If a CRL server or OCSP responder is not available, SSL and S/MIME evaluations could hang or fail.
- **Require for all certs**: This setting requires successful validation of all certificates.

  Use only in a tightly controlled environment that guarantees the presence of a CRL server or OCSP responder. If a CRL server or OCSP responder is not available, SSL and S/MIME evaluations could hang or fail.
- **Local Responder**: If you choose to validate the certificate via OCSP, you can specify a local responder to override that provided in the certificates.
- **Priority**: The priority determines which method (OCSP or CRL) is attempted first.

  If the first method chosen returns a successful validation, the second method is not attempted, unless you choose to require both.

Disable Automatic Login

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Disable automatic login

Description

Disable the automatic login setting. If you enable this group policy, it overrides the Login Options set in the General tab of the Security & Privacy system preference.

For this group policy, clicking Disabled or Not Configured has no effect.

Once enabled, this group policy takes effect when the computer is rebooted.

Disable Location Services

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Disable Location Services

Description

Disable the “Enable Location Services” setting. If you enable this group policy, it overrides the Enable Location Services setting in the Privacy tab of the Security & Privacy system preference.

> **Note**: As of MacOS Catalina, it is not enough to wait for the next group policy refresh or execute adgpupdate. You also need to restart the Mac for this GP to take effect.

For this group policy, clicking Disabled or Not Configured has no effect.

Once enabled, this group policy takes effect at the next group policy refresh interval.

Enable Smart Card Support

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Enable smart card support

Description

Enable users to logon with smart cards. If you enable this group policy, it adds smart card support to the authorization database on Mac computers that are linked to the group policy object.

Delinea smart card support for macOS is based on the macOS modern native framework, CryptoTokenKit

See Configuring a Mac Computer for Smart Card Login for details.

Select **Enable smart card support for the SUDO command**, then when executing the SUDO command, smart card user can authenticate identity by smart card PIN.

Select **Enable smart card support for the SU command**, then when executing the SU command, smart card user can authenticate identity by smart card PIN.

Select **Enable smart card support for the LOGIN command**, then when executing the LOGIN command, smart card user can authenticate identity by smart card PIN.

Select **Enforce smart card login**, then only smart card users with a smart card can log in to the Mac machine.

Edit **Exception group** to add a exception group for the "Enforce smart card login", then any users belong to this group always can log in to the Mac machine by a username and password. In general, we recommend set a exception group, for example, admin, when **Enforce smart card login** is selected.

Select one of options in **Certificate trust behavior** to set smart card certificate trust behavior, the meaning of number:

0: Smart card certificate trust isn't required.

1: Smart card certificate and chain must be trusted.

2: Certificate and chain must be trusted and not receive a revoked status.

3: Certificate and chain must be trusted and revocation status is returned valid.

Once enabled, this policy takes effect dynamically at the next group policy refresh interval.

Enable FileVault 2

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Enable FileVault 2

Description

This group policy allows you to select whether to use one institutional key for multiple Mac computers, or computer-specific ("personal") keys.

To use one institutional key for multiple Mac computers, select **Use Institutional Recovery Key**. Then click **Select** to select the certificate that contains the FileVault master keychain that can unlock the encrypted disk. You must already have created a FileVault master keychain and exported the certificate for the master keychain to a Windows domain server before you perform this step.

To use computer-specific ("personal") keys instead of one institutional key, leave **Use Institutional Recovery Key** unchecked. In this situation, a personal recovery key is created for the Mac computer and stored in the computer object in Active Directory. The key is created and sent to the computer object in Active Directory after the "Managed By" user logs in, logs out, and provides the user password.

This policy is available only for OS X 10.9 and later.

For complete instructions, see Configuring Filevault 2.

> **Note**: Enabling this group policy does not immediately enable FileVault 2 protection on a Mac computer. FileVault 2 protection is enabled when the FileVault-enabled user (that is, the "Managed By" user) logs on to the computer. Disabling this group policy does not disable FileVault 2 protection — disabling FileVault 2 can only be done manually.

Once enabled, this group policy takes effect at the next group policy update interval or when you execute the `adgpupdate` command.

Enable Gatekeeper

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Enable Gatekeeper

Description

Enable the Gatekeeper feature, which controls access to the Mac App store. This policy overrides the "Allow applications downloaded from" setting on the General tab of the Security & Privacy system preference pane.

After enabling the policy, select one of the following options:

- **Mac App Store** Only allow installation of applications that have been downloaded from the Mac App store.
- **Mac App Store** and identified developers. Only allow installation of applications that have been downloaded from the Mac App Store or were created by Apple-sanctioned developers.
- **Anywhere** Allow installation of any applications.

Enable Keychain Synchronization

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Enable Keychain synchronization

Description

This group policy controls whether to enable keychain synchronization, which synchronizes the login keychain to the login user's AD password when a password change is detected.

> **Note**: Keychain synchronization is password-focused and should not be used in smart card environments.

Set the **Password change detection interval (minutes)** option to determine the time (in minutes) between checking for changed passwords. There is a random zero to five minute variance in the actual interval each device is checked for a changed password to maintain performance. As a result, the minimum interval is five minutes.

The default value is 30 minutes.

The **Store AD password in the login Keychain** option is used to streamline updates of the user's login Keychain password. If this option is enabled the Keychain Sync utility stores the user's AD password in the login keychain the next time the user logs in. If the password is changed after the policy is enabled but before the previous password is stored in the login keychain, the keychain sync application requests the previous password.

When this option is selected, the user's AD password is encrypted using a static AES256 key that is unique to that user and stored in the login Keychain as an application password. The key and password are added to the keychain using the SecItemAdd API. In addition, an Access Control List ensures that only the Keychain Sync utility can access the key used to encrypt and decrypt the password.

Delinea recommends disabling Auto Generate New Login Keychain before enabling this policy

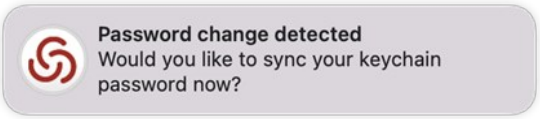
Please note the following limitations with the **Store AD Password in the login Keychain** option:

- This option only works on macOS 10.12 or later.
- The user's AD password is inaccessible when the login keychain is locked.

  The most common scenario that causes this is if a user's AD password is changed and the user logs out before synchronizing the keychain, then logs back in. When the user logs back in, the password check fails due to the new password, locking the login Keychain and preventing the Keychain Sync utility from accessing it.
- Password changes can only be detected when the machine is in connected mode.

User experience when the AD password is already stored in the login Keychain

1. The login user receives a password change notification when his/her password is changed remotely.

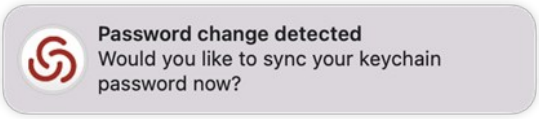


2. When the user clicks **Yes** on the notification, the Delinea Keychain Sync utility appears and asks for the current password to synchronize the keychain.

After entering the current password and clicking **OK**, the Keychain Sync utility synchronizes the login keychain with the new password.

User experience when the AD password is not yet stored in the login Keychain

1. The login user receives a password change notification when his/her password is changed remotely.

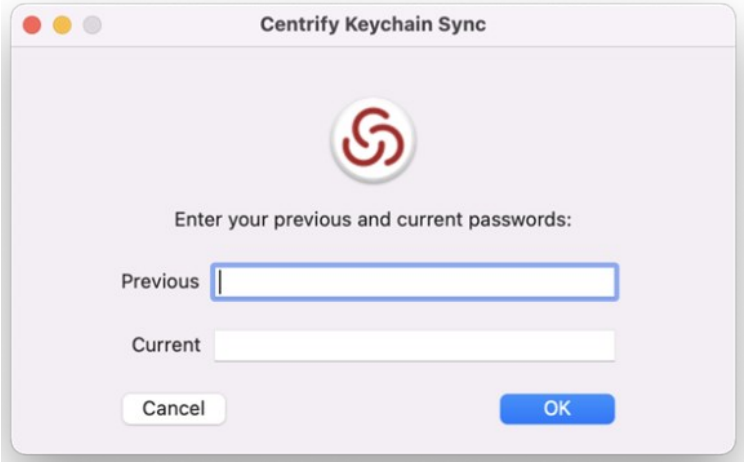


2. When the user clicks **Yes** on the notification, the Delinea Keychain Sync utility appears and asks if the user remembers the previous password.

3. The user clicks **Yes** or **No**.

- If the user clicks **No**, the Keychain Sync utility creates a new login keychain.

- If the user clicks **Yes**, the Keychain Sync utility asks for the previous and current passwords.



After entering the previous and current passwords and clicking **OK**, the Keychain Sync utility synchronizes the login keychain with the new password.

Log Out After Number of Minutes of Inactivity

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Log out after number of minutes of inactivity

Description

Specify the number of minutes of inactivity to allow on a computer before automatically logging out the current user. The default value is 5 minutes.

Setting the value to less than 5 minutes disables automatic logout. If you plan to disable automatic logout, it is recommended that you set the value to 0 to preserve backward compatibility.

> **Note**: Disabling this policy does not disable automatic logout.

This policy takes effect when users log out and log back in after the next group policy refresh.

Require a Password to Wake this Computer from Sleep or Screen Saver

##Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Require a password to wake this computer from sleep or screen saver

Note: This group policy only supports macOS 10.15 and lower.

##Description

Lock the computer screen when the computer goes into sleep or screen saver mode and requires users to enter a user name and password to unlock the screen.

Enabling this group policy is the same as clicking the Require a password to wake this computer from sleep or screen saver option in the Security system preference.

After this group policy is enabled, it takes effect dynamically at the next group policy refresh interval.

Require Password to Unlock Each Secure System Preference

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Require password to unlock each secure system preference

Description

Lock sensitive system preferences to prevent users who aren't administrators from changing them. This group policy requires users to provide an administrator's password to unlock each secure system preference before they can make changes.

If you enable this policy, users must provide an administrator password to access any secure system preference. If the current user is logged on as an administrator and this policy is not configured or disabled, the user can access and change secure system preferences without providing the administrator password.

This policy can take effect dynamically at the next group policy refresh interval.

Use Secure Virtual Memory

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Use secure virtual memory

Description

Prevent passwords from being recoverable from virtual memory.

Any time a password is entered, it is possible for system to write that password in a block of memory that it dumps to a file in /var/vm, making the password recoverable.

Enabling this group policy ensures that the virtual memory /var/vm files are encrypted, preventing any passwords written there from being recovered.

This policy can take effect dynamically at the next group policy refresh interval.

Allow All Applications to Access the Auto-Enrollment Private Key(S)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Allow all applications to access the auto-enrollment private key(s)

Description

Enabling this policy allows all applications to access the auto-enrollment private key(s) in the System keychain.

See Configuring Auto-Enrollment for more information about auto-enrollment keys.

Note: This setting only applies to a new auto-enrollment private key(s); it will not update already imported auto-enrollment private key(s) that are in the System keychain.

Allow Specific Applications to Access the Auto-Enrollment Private Key(S)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Allow specific applications to use the auto-enrollment private key(s)

Description

Enabling this policy allows specified applications to access the auto-enrollment private key(s) in System keychain.

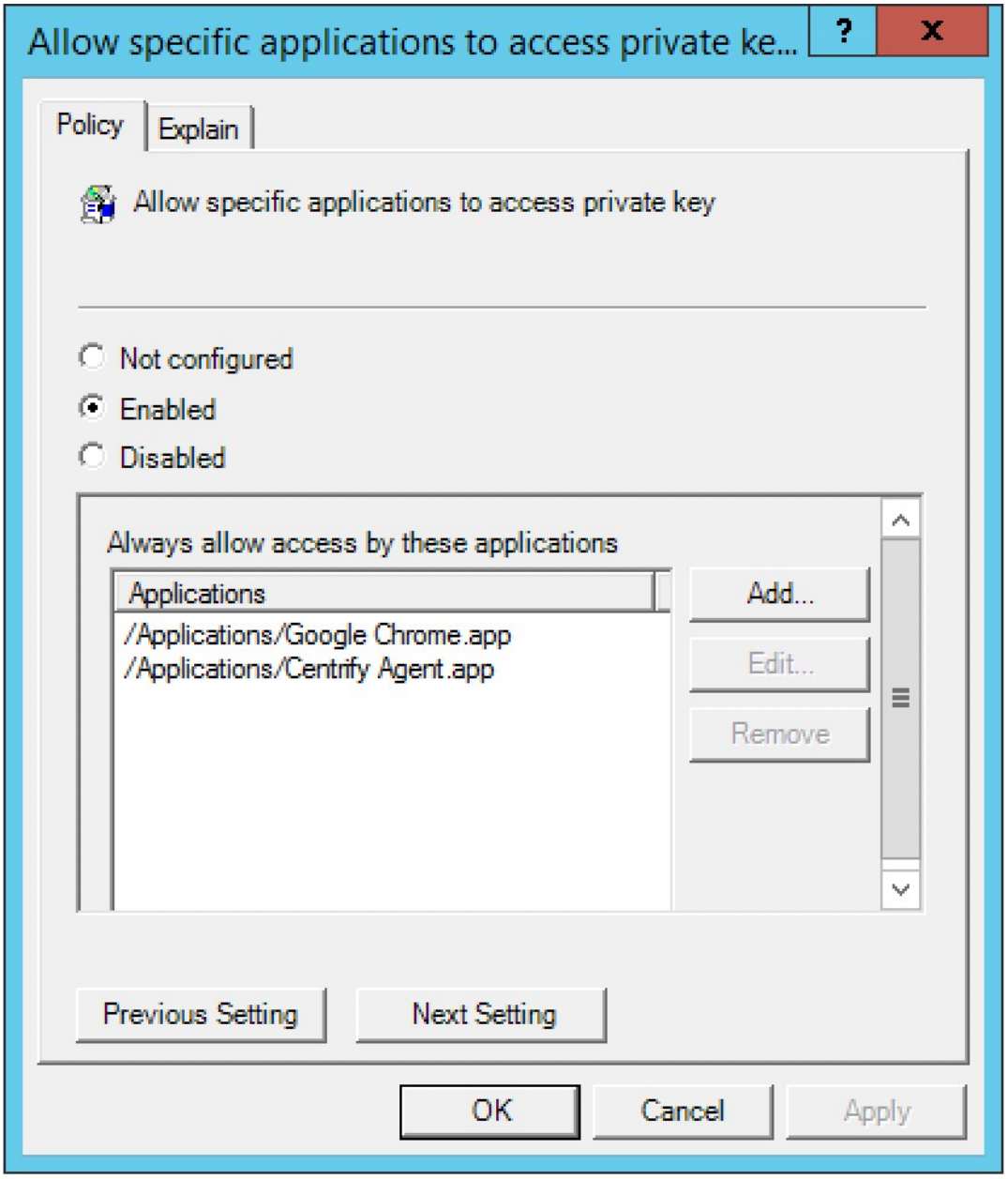
After you enable this policy, click **Add** to enter the path to the application you want to allow access to the auto-enrollment private key, then click **OK**. You can click **Add** again to add additional applications.

For example, to give Google Chrome and Delinea Agent access to the auto-enrollment private key, enter the application path for Google Chrome:

```
/Applications/Google Chrome.app
```

Click **OK**. Then click **Add** and enter the application path for Delinea Agent:

```
/Applications/Centrify Agent.app
```



After this group policy is enabled, the list of applications specified in the group policy are added to the access control list of the auto-enrollment private key in System keychain.

See Configuring Auto-Enrollment for more information about auto-enrollment keys.

> **Note**: This setting only applies to a new auto-enrollment private key. It does not change auto-enrolled private keys that are already in the keychain.

If the group policy **Allow all applications to access the auto-enrollment private key(s)** (above) is enabled, this group policy will be ignored.

Do Not Allow the Private Key(S) to be Extractable

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Do not allow private key(s) to be extractable

Description

Enabling this policy prevents exporting the auto-enrollment private key(s).

> **Note**: This setting only applies to a new auto-enrollment private key. It does not change the auto-enrolled private key(s) that are already in the keychain.

Store The Private and Public Key(S) Only in the Keychain

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Store the private and public key(s) only in the keychain

Description

Enable this group policy to store the auto-enrollment key(s) only in the keychain.

User certificate auto-enrollment always uses the Keychain and is not controlled by any Group Policy.

> **Note**: 802.1X profiles installed through the "Mac OSX Settings -> 802.1X Settings" Group Policies will no longer be signed if this GP is enabled before profiles are installed.

Services

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services** settings to manage access to the service options from the Sharing () system preference on Mac computers. These group policies correspond to the options displayed on the Services pane. For example:

Enable Personal File Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Personal File Sharing

Description

Allow users on other Mac computers access to Public folders on the local computer. If you enable this group policy, all users can access files in the Public folder through the Apple File Sharing protocol. Users with appropriate permission can also access other folders on the local computer if properly authenticated.

Enabling this group policy is the same as opening the Sharing system preference, selecting **File Sharing**, then clicking the **Options** button and selecting the **Share Files and Folders using AFP** option.

On Mac OS X Servers, enabling this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Windows Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Windows Sharing

Description

Allow users on Windows computers access to shared folders on the local computer through SMB/CIFS file shares.

Enabling this group policy is the same as opening the Sharing system preference, selecting **File Sharing**, then clicking the **Options** button and selecting the **Share Files and Folders using SMB** option.

On Mac OS X Servers, this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Personal Web Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Personal Web Sharing

Description

Allow users on other computers to view Web pages in each user's sites folder on the local computer.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Web Sharing option.

On Mac OS X Servers, enabling this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Remote Login

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Remote Login

Description

Allow users on other computers to access this computer using SSH.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Remote Login option.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable FTP Access (deprecated)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable FTP Access

Description

Allow users on other computers to exchange files with this computer using FTP applications.

Enabling this group policy is the same as opening the Sharing system preference, selecting **File Sharing**, then clicking the **Options** button and selecting the **Share Files and Folders using FTP** option.

On Mac OS X Servers enabling this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Apple Remote Desktop

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Apple Remote Desktop

Description

Allow others to access this computer using the Apple Remote Desktop program.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Remote Management option.

If you enable this group policy, you can set the following access privileges:

- Allow guest users to request permission to control the screen
- Prevent VNC viewers from controlling the screen.

Because allowing VNC viewers to control the screen requires setting a password to take control of the screen and this behavior presents a potential security issue, this group policy can only be used to disallow VNC access.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Remote Apple Events

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Remote Apple Events

Description

Allow applications on other Mac computers to send Apple Events to the local computer.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Remote Apple Events option.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Printer Sharing

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Printer Sharing

Description

Allow other people to use printers connected to the local computer.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Printer Sharing option.

On Mac OS X Servers, enabling this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Enable Xgrid

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Services > Enable Xgrid

Description

Allow clustered Mac OS Xgrid controllers to distribute tasks to the local computer for completion.

Enabling this group policy is the same as opening the Sharing system preference and selecting the Xgrid Sharing option.

On Mac OS X Servers enabling this policy has no effect.

This policy can take effect dynamically at the next group policy refresh interval without rebooting the computer.

Software Update Settings

Path

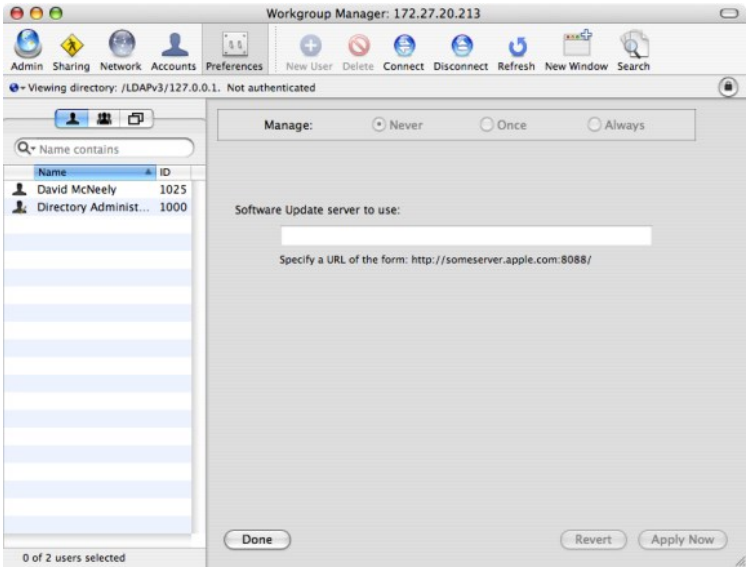
Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Software Update Settings

Description

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Software Update Settings** group policies to manage software updates. The group policies in this category enable you to set the interval for checking for software updates and to identify a specific server from which updates should be received.

These group policies correspond to settings you make using the Software Update ( ) system preference on client Mac OS X computers and the Software Update preference in the Workgroup Manager on Mac OS X servers. For example, the interval for checking for software updates is typically configured Software Update system preference on client Mac computers:

Note: Identifying a software update server to use for downloading updates is configured on a Mac OS X server using the Software Update preference in the Workgroup Manager. For example:



The software update group policies are computer policies, applied as the root user, and apply to all users of the computer. Setting these group policies updates the plist files for individual users with the group policy parameters, such as update server URL, update interval, and so on. However, to prevent local users from using Software Update in System Preferences to manually set software update server parameters, an administrator should also limit access to the Software Update Preferences Pane by setting the group policy, **Limit Items Shown in System Preferences**, and then enabling the group policy, **Enable System Preferences Pane: System** › **Enable Software Update**.

Otherwise, you may see anomalous behavior. For example, a user can open Software Update and change parameters, such as disabling software updates (by deselecting Check for updates). If the user then re-enables software updates, the update server resets to the Apple software update server, not the server specified in the software update server group policy. However, at the next login, or at the next adgpupdate period, the Server URL and other group parameters will be re-applied.

The Software Update Settings contain separate folders that allow you to specify a different update server for each operating system version that you are running. For example, if you have computers with different versions of OS X in your environment, you can specify a different update server for each one by enabling the Specify software update server policy in each of the version-specific folders. In order to do this you must enable Use version specific settings.

If you do not enable Use version specific settings, Legacy Settings are used instead. If you applied Software Update Settings to computers running previous versions of the product, those settings are in Legacy Settings, though you may update them if you wish.

Note: The Automatically download and install software updates policy applies to all computers, regardless of version.

Automatically Check For Software Updates (Legacy, Currently Supported)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Software Update Settings > Automatically check for software updates (Legacy, Currently supported)

Description

Note: There are actually separate versions of this policy in version-specific folders.

Periodically check for updated versions of the software installed on the local computer and automatically download and install newer versions. You can configure the version-specific versions of this policy the same way you can configure the Software Update system preference for the corresponding operating system version.

This policy takes effect when users log out and log back in.

Use Version Specific Settings

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Software Update Settings > Use version specific settings

Description

Enable the use of version-specific settings.

You can then set platform-specific preferences settings for each platform in your environment, which enables you to specify a different update server depending on the version of Mac OS X running on a computer. For example, if you have only 10.10 computers, you can enable this policy and then use Mac OS X 10.10 settings. If you have 10.10 and 10.9 computers, enable this policy, and then configure the version-specific policies as appropriate:

- Mac OS X 10.10 Settings
- Mac OS X 10.9 Settings

If this policy is disabled or not configured, Legacy Settings are used instead of version-specific settings. Likewise, Delinea versions prior to 4.4.2 always use Legacy Settings and ignore this policy setting.

If you configured Software Update Settings with a version of the product prior to 4.4.2, these settings are saved to Legacy Settings when you upgrade to the current Delinea version. You can keep or edit these settings as you wish.

Specify Software Update Server (Legacy, Currently Supported)

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Software Update Settings > Specify software update server (Legacy, Currently supported)

Description

Note: There are actually separate versions of this policy in version-specific folders.

This enables you to specify a separate update server based on the version of the Mac OS X computer.

Type the URL that identifies the computer you are using as the software update server. It is recommended that you specify the hostname of the server rather than the IP address; for example:

```
http://myHost.local:8088
```

In addition, to ensure that DNS associates the hostname of the update server with the IP address, add a line such as the following to the `/etc/hosts` file:

```
192.168.2.79 myHost.local
```

where: `192.168.2.79` is the IP address of the update server and `myHost.local` is the hostname.

This policy can take effect dynamically at the next group policy refresh interval.

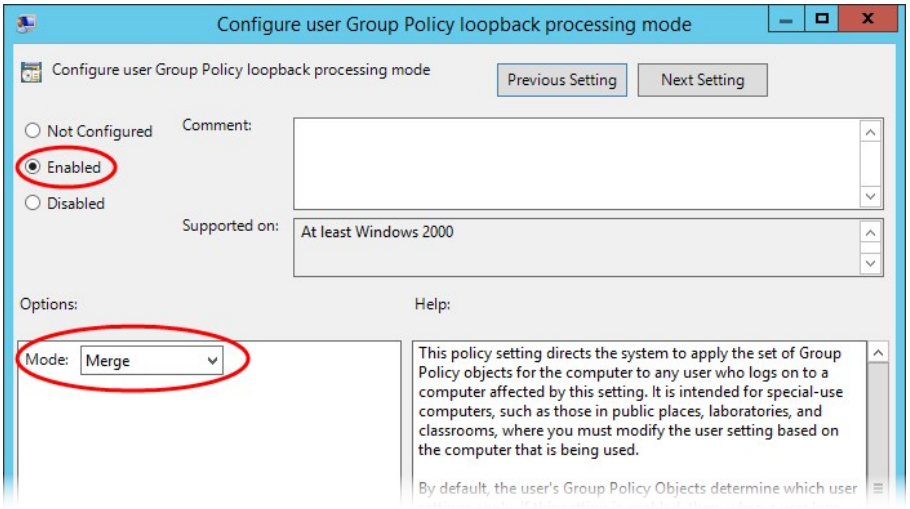
Centrify group policies allow administrators to extend the configuration management capabilities of Windows Group Policy Objects to managed Mac computers and to users who log on to Mac computers. This chapter describes the Mac group policies that can be applied to Mac users

The user-based group policies are defined in the Centrify Mac administrative template (centrify_mac_settings.xml) and accessed from **User Configuration > Policies > Centrify Settings > Mac OS X Settings**.

Group policies are only applied to users and computers in the GPO's linked OU and any child OUs. If your users and computers are in different OUs (which is common), Delinea recommends using user Group Policy loopback processing to make sure user policies are applied to everyone who logs on to a Mac. This is a standard Microsoft Group Policy that applies to every user to the computer.

To implement user Group Policy loopback processing mode

1. In the Group Policy Management Editor, navigate to **Computer Configuration > Administrative Templates > System > Group Policy > Configure user Group Policy loopback processing mode**.
2. Enable the policy, set **Mode:** to **Merge**, then click **OK**.



See https://technet.microsoft.com/en-us/library/cc978513.aspx for more information about loopback processing.

See Understanding Group Policies for Mac Users and Computers for general information about how to use group policies to manage Mac settings and for information on how to install the group policy administrative templates.

> **Note**: For additional information about creating and using group policies and Group Policy Objects, see your Windows or Active Directory documentation. For more information about adding and using other Centrify group policies that are not specific to Mac computers and users, see the *Group Policy Guide*.

Setting User-Based Policies

This section describes user-based policies for Mac that you can set. The following table provides a summary of the group policies you can set for Mac users. These group policies are in the Centrify Mac administrative template (centrify_mac_settings.xml) and accessed from **User Configuration > Policies > Centrify Settings > Mac OS X Settings**.

> **Note**: Group policies are only applied to users and computers in the GPO's linked OU and any child OUs. Enable **Computer Configuration > Administrative Templates > System > Group Policy > Configure user Group Policy loopback processing mode** in Merge mode to make sure user policies are applied to everyone who logs on to a Mac.

802.1X Wireless Settings	Create user profiles for wireless authentication. This group policy corresponds to 802.1X Options in the **Networks** system preference.

Application Access Settings (deprecated)	Control the specific applications users are either permitted to use or prohibited from using. These group policies correspond to **Applications** preferences set in the Workgroup Manager.
Desktop Settings	Control the desktop and screen saver options for users on Mac computers. These group policies correspond to settings in the **Desktop & Screen Saver** system preference.
Dock Settings	Control the look and operation of the Dock displayed on the user's desktop. These group policies correspond to **Dock** preferences set in the Workgroup Manager.
Finder Settings	Specify whether to use the standard Finder, or the Simple Finder, which restricts users to applications and folders in the Dock.
Folder Redirection	Redirect specified network home folders to the local computer to improve performance.
Import Settings	Specify plist files to import preferences from another computer. This group policy corresponds to the import plist functionality in Workgroup Manager.
Login Settings	Specify frequently used applications, folders, and server connections to open when a user logs in. This group policy corresponds to the login functionality in Workgroup Manager.
Media Access Settings	Control the specific media types users are either permitted to use or prohibited from using. These group policies correspond to **Media Access** preferences set in the Workgroup Manager.
Mobility Settings	Control the synchronization rules applied for users access services from mobile devices. These group policies correspond to **Mobility** preferences set in the Workgroup Manager.
Scripts (Login/Logout)	Specify login and logout scripts that run when Active Directory users log on or log out.
Security & Privacy Settings	Control the secure login options for users on Mac computers. These group policies correspond to settings in the **Security** system preference.
System Preference Settings	Control the specific system preferences displayed for users. These group policies correspond to **System Preferences** set in the Workgroup Manager.

802.1X Wireless Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X** settings to create profiles for wireless network authentication. The profiles you specify with these group policies are created in the Network system preferences pane.

Specify User Profiles (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > 802.1X Wireless Settings > Specify User Profiles (Deprecated)

Description

Enable this policy to specify 802.1X User Profiles for wireless network authentication.

When using a user profile, a user will be prompted for username and password to authenticate to a wireless network after login.

To add a user profile

1. Enable the policy and click **Add** to enter the profile name and setting.
2. Type a name for the profile.
3. Type the setting using the following format:
    - Network;Security Type;Authentication Method, where each field is separated by a semi-colon ;.
    - Network is the wireless network name
    - Security type is one of 802.1X WEP, WPAEnterprise, WPA2 Enterprise
    - Authentication method is one or more of the following, separated by commas: TTLS, PEAP, TLS, EAP-FAST, LEAP, MD5

For example:

OFFICE1;WPA Enterprise;PEAP

OFFICE2;802.1X WEP;TTLS,PEAP

Set the **Automatically turn on Airport** option to automatically turn on AirPort device if this type of profile is specified. Otherwise, the status of the AirPort device will not change.

Once enabled, this policy takes effect dynamically at the next group policy refresh interval.

Application Access Settings (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings** group policies to manage the applications Mac users are allowed to open or prevented from opening.

These group policies correspond to settings you can make using the Applications preference in the Workgroup Manager.

Permit/Prohibit Access to Application List: Applescript (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to application list: AppleScript

Description

Select the specific applications in the Finder's Applications/AppleScript folder that users are permitted to use if you selected **Users can only open these applications**, or not allowed to use if you selected **Users can open all applications except these**.

This policy is only effective if the **Permit/prohibit access to applications** group policy is enabled. If the **Permit/prohibit access to applications** group policy is not configured or disabled, this group policy is ignored.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Permit/Prohibit Access to Application List: Applications (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to application list: Applications

Description

Select the specific applications in the Finder's Applications folder that users are permitted to use if you selected **Users can only open these applications**, or not allowed to use if you selected **Users can open all applications except these**.

This policy is only effective if the **Permit/prohibit access to applications** group policy is enabled. If the **Permit/prohibit access to applications** group policy is not configured or disabled, this group policy is ignored.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Permit/Prohibit Access to Application List: Server (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to application list: Server

Description

Select the specific applications in the Finder's Applications/Server folder that users are permitted to use if you selected **Users can only open these applications**, or not allowed to use if you selected **Users can open all applications except these**.

This policy is only effective if the **Permit/prohibit access to applications** group policy is enabled. If the **Permit/prohibit access to applications** group policy is not configured or disabled, this group policy is ignored. In addition, this policy is only applicable for Mac OS X Server computers.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Permit/Prohibit Accesst to Application List: Utilities (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to application list: Utilities

Description

Select the specific applications in the Finder's Applications/Utilities folder that users are permitted to use if you selected **Users can only open these applications**, or not allowed to use if you selected **Users can open all applications except these**.

This policy is only effective if the **Permit/prohibit access to applications** group policy is enabled. If the **Permit/prohibit access to applications** group policy is not configured or disabled, this group policy is ignored.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Permit/Prohibit Access to Applications (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to applications

Description

Allow other policies to specify the applications that users are permitted to access or prohibited from accessing. You must enable this policy for any other application access group policies to take effect. Once enabled, only the applications explicitly specified in Application List policies are permitted or prohibited.

If you enable this policy, in **Access mode**, select one of the following:

- **Users can only open these applications** to grant access only to the applications you select with the other application access policies.

  Note: If you select the option, **User can also open all applications on local volumes**, users can access any local applications. Restrictions only apply to applications on CDs, DVDs, or external disks.

- **Users can open all applications except these** to prevent access only to the applications you select with the other application access policies.

You can also set the following options in this group policy:

- Select **User can also open all applications on local volumes** to allow access to applications on a computer's local hard drive.

  If selected, users can access any local applications in addition to the applications explicitly approved using the other application access policies. If you uncheck this option, users can only access applications on CDs, DVDs, or external disks that have been explicitly approved.

- Select **Allow approved applications to launch non-approved applications** to allow approved applications to open applications that aren't explicitly approved.

  For example, if users click a link in an email message, this option allows the email application to open a browser to display the Web page even if the browser is not listed as an approved application. To prevent approved applications from opening applications that aren't explicitly approved, uncheck

this option.

- Select **Allow UNIX tools to run** to allow applications or the operating system to run tools, such as the QuickTime Image Converter, without explicitly listing them as approved applications.

  These tools usually operate in the background, but can be run from the command line. If you want to prevent access to these tools, do not check this option.

Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Permit/Prohibit Access to the User-Specific Applications (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Application Access Settings > Permit/prohibit access to the user-specific applications

Description

Define a list of additional applications that users are permitted to run if you selected **Users can only open these applications**, or not allowed to use if you selected **Users can open all applications except these**. If enabled, you must specify the CFBundleIdentifier to identify the application; for example, for the Firefox browser, the CFBundleIdentifier is: org.mozilla.firefox. To find the CFBundleIdentifier complete the following steps:

1. In the Finder, locate the application to control.
2. Control-click or right-click the application, then select **Show Package Contents**.
3. If necessary, expand the **Contents** folder, then open info.plist with a text editor.
4. Find the string: CFBundleIdentifier.

   On the next line is the application's CFBundleIdentifier; for example:

```
org.mozilla.firefox
```

1. Use org.mozilla.firefox to identify the Firefox browser.

To add an application to the list, select **Enabled**, then click **Add** and enter the CFBundleIdentifier and click **OK**.

You may also control access to system preference panes by using the CFBundleIdentifier. You can find the CFBundleIdentifier for system preference panes in /System/Library/PreferencePanes. You can specify any application object that has a CFBundleIdentifier in its info.plist file.

> **Note**: Some applications may not have a CFBundleIdentifier (when you right-click the application name, there is no **Show Package Contents** menu item). In this case, you cannot add the application to the list of permitted or prohibited applications.

This policy is only effective if the **Permit/prohibit access to applications** group policy is enabled. If the **Permit/prohibit access to applications** group policy is not configured or disabled, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

Automount Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Automount Settings

Description

Use the Automount Settings to automatically mount network shares and the user's Windows home directory when a user logs in.

Automount Network Shares

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Automount Settings > Automount network shares

Description

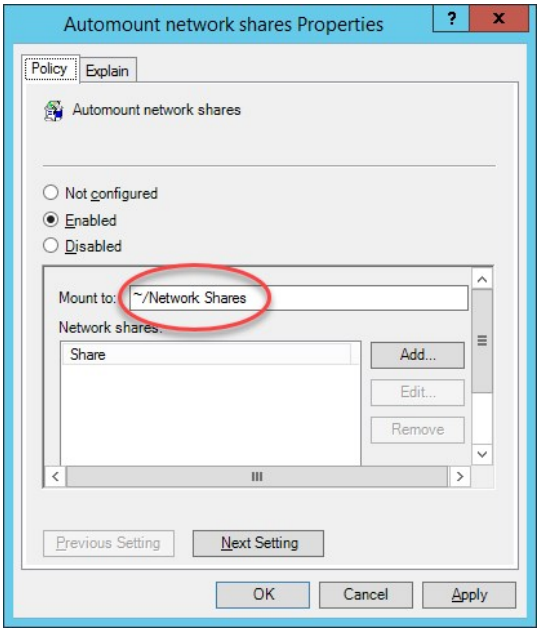
Specify the network shares to automatically mount when a user logs in. The default network share mount location is *User_Home*/Network Share.

This policy supports SMB, AFP, and NFS shares.

To add a share

1. Enter a path to mount the share in the **Mount to:** field.

   The path should start with / or ~. The default value is ~/Network Shares. In this case, network share folders would be mounted under the directory Network Shares of user's home directory.



2. Click **Enabled**, then click **Add** and enter the share in one of the following formats:

keyword://server/share

where:

- keyword is one of smb, nfs, afp
- server is the name or IP address of the server and can include a user or user and password in the form: user:@server or user:password@server.
- share can include spaces and be followed by a subdirectory.

For example, the following are all valid share specifications:

smb://acme.com/MacUsers

smb://acme.com/Mac Users

smb://acme.com/MacUsers/Shared_resources

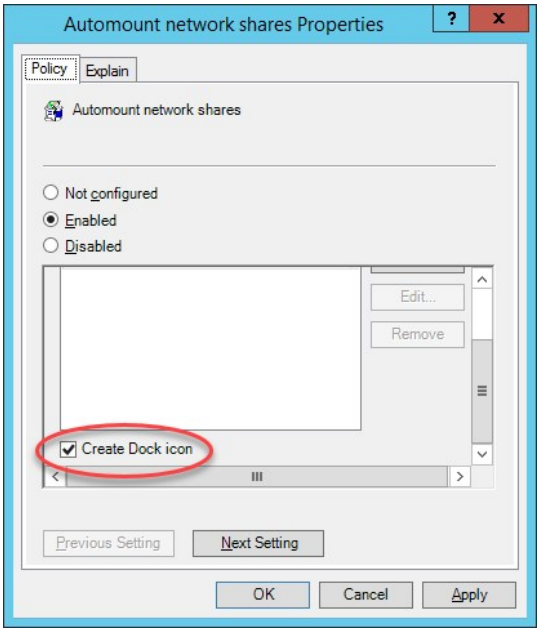
smb://jsmith:pass1234@acme.com/MacUsers

afp://acme.com/Users_server

nfs://acme.com/MacUsers

nfs://192.168.0.1/MacUsers

1. (Optional) Select **Create Dock icon** to create a link to the network share in the user's Dock.



Once enabled, this policy takes effect when a user logs out and back in to a computer.

Automount User's Windows Home

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Automount Settings > Automount user's Windows home

Description

Automatically mount the user's Windows home directory when the user logs in.

Specify the Windows home directory by using the Profile tab for a user in Active Directory Users and Computers (ADUC).

Once enabled, this policy takes effect when a user logs out and back in to a computer.

Create Alias Instead of Symbolic Link

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Automount Settings > Create alias instead of symbolic link

Description

This group policy is provided for compatibility with Delinea releases earlier than 2015. If you are using release 2015 or later, do not use this group policy.

In releases prior to 2015, the default mount point for network shares was /var/centrify/mnt/user. Starting with release 2015, the default mount point for network shares is User_Home/Network Share.

In Delinea releases prior to 2015, the "Automount network shares" group policy creates symbolic links to the specified shared network directories. However, certain versions of Microsoft Office are unable to save files to a shared folder by using the symbolic link (the link is greyed-out). The "Create alias instead of symbolic link" group policy corrects the problem by creating an alias instead of a symbolic link. In release 2015 or later, because of the new mount location, symbolic links are not required, and this group policy has no effect.

If you enable this group policy, the alias points to network shares that are automatically mounted when a user logs in.

> **Note**: The operating system treats an alias as a file, which means that you cannot use the Terminal program to access files or folders that are pointed to by the alias.

Once enabled, this policy takes effect when a user logs out and back in to a computer.

Custom Settings

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Custom Settings

Description

Use the **User Configuration › Policies › Centrify Settings › Mac OS X Settings › Custom Settings › Install MobileConfig Profiles** group policy to install mobile configuration profiles. This policy installs a user profile. To install a device profile, use the same policy in **Computer Configuration › Centrify Settings › Mac OS X Settings › Custom Settings.**

Install MobileConfig Profiles

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Custom Settings › Install MobileConfig profiles

Description

Enable this group policy to install mobile configuration profiles on managed Mac computers.

> **Note**: There is a Computer Configuration version of this policy (which installs device profiles) and a User Configuration version (which installs user profiles).

> **Note**: This group policy only supports macOS 10.15 and lower.

Before enabling this policy, you must create a directory and copy mobile configuration files to SYSVOL on the domain controller. SYSVOL is a well-known shared directory on the domain computer that stores server copies of public files that must be shared throughout the domain.

Specifically, create the following directory on the domain controller: `\\domainName\SYSVOL\domainName\mobileconfig`

and copy one or more mobile configuration profile files to this directory. See Deploy Configuration Profiles to Multiple Computers for details on how to do this.

To specify mobile configuration files to install, enable the policy, then click **Add**. Enter the name of a mobile configuration file that you placed in SYSVOL on the domain controller. Include the `.mobileconfig` suffix with the name.

If you specify a file that is not in the SYSVOL mobileconfig directory, the profile will not be installed.

If you add new files to the existing list in the group policy, those profiles will be installed — existing profiles will not be touched. If you remove previously specified files, the profiles defined by these files will be uninstalled.

If you add two or more profile files that have the same `payloadIdentifier`, only one of them will be installed.

If you change the group policy to "Disabled" or "Not Configured", all existing profiles that were installed previously by the group policy will now be uninstalled from the managed Mac computers.

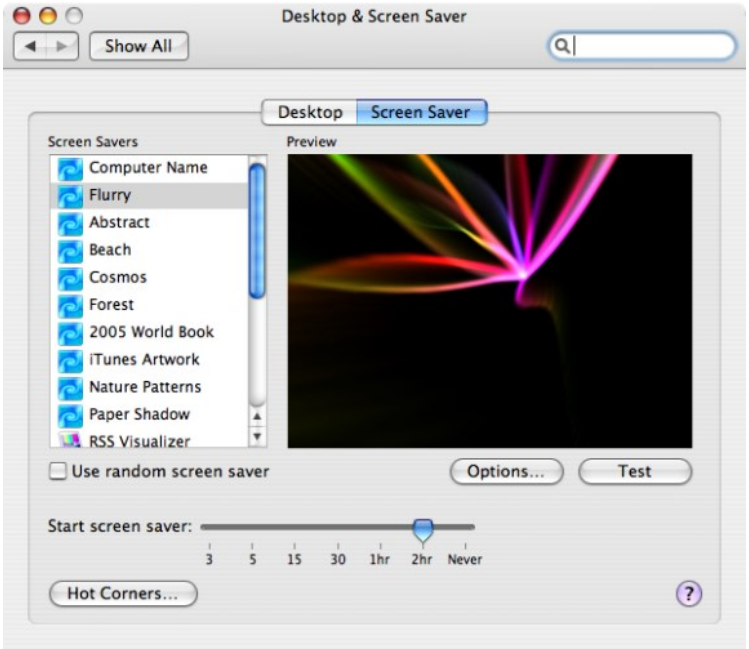
Desktop Settings

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Desktop Settings

Description

Use the **User Configuration › Policies › Centrify Settings › Mac OS X Settings › Desktop Settings** group policy to manage the start time for the screen saver from the Desktop & Screen Saver () system preference on Mac computers. This group policy corresponds to the **Start screen saver** option displayed on the Screen Saver pane. For example:



Set Computer Idle Time for Starting Screen Saver

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Desktop Settings > Set computer idle time for starting screen saver

Description

Select the length of time to wait before starting the screen saver. If you enable this group policy, you can specify the number of minutes to wait while a computer is not in use before starting the screen saver. For example, if you want the screen saver to start after a computer has been idle for 10 minutes, you can set Start screen saver to 10 minutes.

Disabling this policy does *not* disable the screen saver. To disable the screen saver, enable this policy and set the value to 0.

Although you may specify values greater than 60 minutes, and the screen saver works appropriately, the Macintosh Screen Saver dialog box shows values that are greater than 60 as **Never**.

Enabling this group policy is the same as selecting when to start the screen saver using the **Start screen saver** slider in the Desktop & Screen Saver system preference.

Once enabled, this group policy takes effect when users log out and log back in.

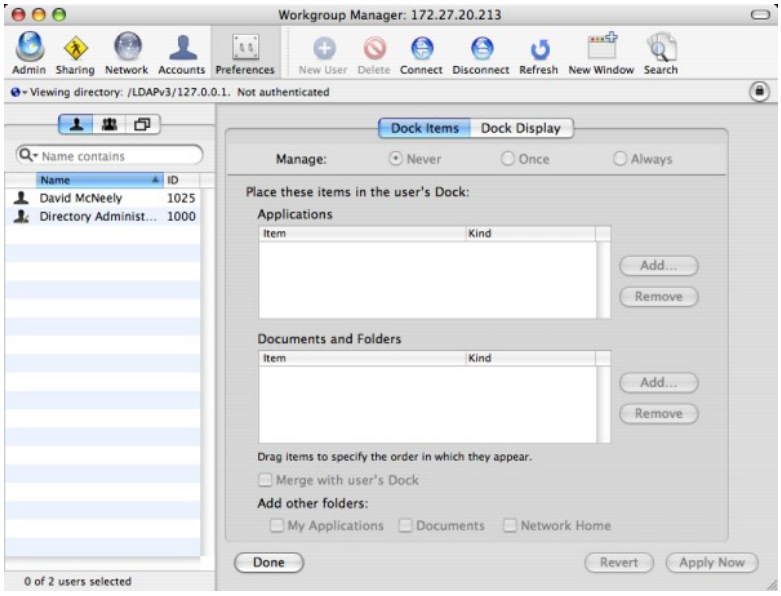
Dock Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings** group policies to manage the characteristics of the Dock for Mac users. These settings correspond to the Dock preferences you can manage using the Workgroup Manager. In the Workgroup Manager, the Dock Items pane controls the items placed in the Dock and whether the workgroup Dock is merged with the user's Dock, and the Dock Display pane controls attributes such as the Dock size, magnification, position, and animation. For example:



Add Other Folders to the Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Add other folders to the Dock

Description

Add icons for the other commonly-used folders to the Dock. You can choose to add the following folder icons to the Dock:

- My Applications
- Documents

The **My Applications** folder contains aliases to all approved applications you have defined in the Application list. If you do not manage access to applications, all available applications are included in the My Applications folder. If you enable Simple Finder, you should display the My Applications folder.

The **Documents** folder is the Documents folder found in the user's home folder. For example, the /Users/username/Documents folder for local user accounts.

Once enabled, this group policy takes effect when users log out and log back in.

Adjust the Dock's Icon Size

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Adjust the Dock's icon size

Description

Set the approximate size of Dock icons in pixels. The valid settings for the Dock size range from 16 pixels (small) to 128 pixels (large). The default size is 80 pixels.

> **Note**: This setting is approximate because the actual size of Dock icons depends on screen resolution and the number of icons in the Dock.

Once enabled, this group policy takes effect when users log out and log back in.

Adjust the Dock's Magnified Icon Size

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Adjust the Dock's magnified icon size

Description

Set the level of magnification to use for items in the Dock. If you enable this group policy, icons in the Dock are magnified to display in a larger size as the pointer moves over them. The valid settings for Dock magnification range from 16 pixels for minimum magnification to 128 pixels for maximum magnification. The default size is 80 pixels.

If you do not configure or disable this group policy, icons in the Dock are not magnified when the pointer moves over them.

Once enabled, this group policy takes effect when users log out and log back in.

Adjust the Dock's Position on Screen

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Adjust the Dock's position on screen

Description

Specify the location for displaying the Dock on the screen. If you enable this group policy, you can position the Dock on the left, bottom, or right of the screen. The default location for displaying the Dock is at the bottom of the screen.

Once enabled, this group policy takes effect when users log out and log back in.

Adjust The Effect Shown When Minimizing the Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Adjust the effect shown when minimizing the Dock

Description

Specify the effect to use when a window or application is minimized and placed in the Dock. The valid effects are:

- Genie
- Scale
- Suck

Once enabled, this group policy takes effect when users log out and log back in.

Animate Opening Applications

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Animate opening applications

Description

Animate application icons so that the icon displayed in the Dock bounces when the user opens the application.

Once enabled, this group policy takes effect when users log out and log back in.

Automatically Hide and Show the Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Automatically hide and show the Dock

Description

Hide the Dock from view automatically. If you enable this policy, the Dock is hidden during normal operation. The Dock is then automatically displayed again if the pointer moves over the position on the screen where the Dock is located.

Once enabled, this group policy takes effect when users log out and log back in.

Lock the Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Lock the Dock

Description

Lock the applications displayed in the Dock. If you enable this policy, icons cannot be moved into or out of the Dock.

Once enabled, this group policy takes effect when users log out and log back in.

Place Applications in Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Place applications in Dock

Description

List the applications to include in the Dock. After you enable this policy, click **Add** to enter the path to the application you want included in the Dock. Then click **OK**. You can click **Add** again to add additional applications. For example, to add Firefox and Chess icons to the Dock, type the application paths:

/Applications/Firefox.app

Click **OK**. Then click **Add** and enter:

/Applications/Chess.app

The icons for the applications you specify are placed to the left or above the separator line in the Dock in the order you enter them, up to 10 items. if you add more than 10 the order may be random. If the path to an application is incorrect, a question mark (?) is displayed in the Dock in place of the application's icon.

This group policy does not sort icons from the initial system list. To sort these items, such as the Mail application icon, you can add the item to the list.

Once enabled, this group policy takes effect when users log out and log back in.

Place Documents and Folders in Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Place documents and folders in Dock

Description

List the documents or folders to include in the Dock. After you enable this policy, click **Add** to enter the path to the folder or document you want to include in the Dock. Then click **OK**. You can specify additional folders or documents by clicking **Add** again. For example, to add the Users folder and the Copyright.txt document to the Dock, type the paths to each:

/Users

Click **OK**, then click Add and type:

/Documents/Copyright.txt

The icons for the items you specify are placed to the left or above the separator line in the Dock. Items are sorted in the order you enter them up to 10 items. If you specify more than 10 items the order may be random. If the path to an item is incorrect, a question mark (?) is displayed in the Dock.

> **Note**: You may not specify the path to a network share; for example, smb://serverName. Network share paths are implemented as aliases, which work differently than folder and document paths. If you specify a network share, a question mark (?) is displayed in the Dock.

Once enabled, this group policy takes effect when users log out and log back in.

Merge with User's Dock

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Dock Settings > Merge with user's Dock

Description

Merge the Workgroup Dock settings with the user's Dock.

Once enabled, this group policy takes effect when users log out and log back in.

Finder Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Finder Settings

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Finder Settings** group policies to configure Finder commands, preferences and views.

The **Configure Finder Commands (Deprecated)** policy, below, allows you to control which commands are available in the Apple menu and Finder menus for users.

The **Configure Finder Preferences (Deprecated)** policy, below, enables you to specify the type of Finder for the user environment. After enabling the policy, you can choose one of two types from the drop-down list:

- **Normal Finder** applies the standard Mac desktop. This is the default value, and is the environment that all users will have if the policy is not enabled.
- **Simple Finder** restricts users to applications that are in the Dock.

When Simple Finder is enabled, users cannot open applications, open, modify, or delete documents, or create folders in the Finder. They also cannot mount network drives. They can only use items that are in the Dock. Use the **Dock Settings** policies above to configure the Dock; for example, enable **Place Applications in Dock** and **Place Documents and Folders in Dock** to control the applications and folders that users can access.

The **Configure Finder Preferences (Deprecated)** policy, below, enables you to control the arrangement and appearance of items on the user's desktop, in Finder windows, and in the top-level folder of the computer.

Configure Finder Commands (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Finder Settings > Configure Finder commands (Deprecated)

Description

Specify the commands in Finder menus and the Apple menu that are available to users. Select commands from the following list:

- **Connect to Server**

  Select to allow users to connect to a remote server by choosing 'Connect to Server' in the Finder Go menu. Deselect to prevent users from accessing this command.

- **Go to iDisk**

  Select to allow users to connect to an iDisk by choosing 'Go to iDisk' in the Finder Go menu. Deselect to prevent users from accessing this command.

- **Eject**

  Select to allow users to eject discs (for example, CDs, DVDs, floppy disks, or FireWire drives). Deselect to prevent users from ejecting disks.

- **Burn Disc**

  Select to allow user on computers with relevant hardware to burn discs. Deselect to prevent users from burning disks.

- **Go to Folder**

  Select to allow users to open a specific folder by choosing the 'Go to Folder' command in the Finder Go menu. Deselect to prevent users from using the

'Go to Folder' command.

- **Restart**

  Select to allow users to restart the computer they're using, or deselect to prevent them from restarting the computer.

- **Shut Down**

  Select to allow users to shut down the computer they're using, or deselect to prevent them from shutting down the computer.

Once enabled, this group policy takes effect when users log out and back in.

Configure Finder Preferences (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Finder Settings > Configure Finder preferences (Deprecated)

Description

Configure Finder preferences, including whether to use normal or Simple Finder, which items to show on the desktop, how a new window behaves, and whether to show filename extensions and the Empty Trash warning.

Select from the following options:

- **Finder type**

  Select the normal Finder or Simple Finder as the user environment. The normal Finder looks and acts like the standard Mac desktop. Simple Finder removes the ability to use a Finder window to access applications or modify files, limiting users' access to only what is in the Dock. In addition, users can't mount network volumes, create folders, or delete files.

- **Show these items on the Desktop**

  Choose whether users see icons for local hard disks, external disks, CDs (DVDs and iPods), and connected servers on the desktop.

  If you hide them, icons for disks and servers still appear in the top-level folder when a user clicks the Computer icon in a Finder window's toolbar.

- **New Finder window shows**

  Select **Home** to show items in the user's home folder, or select **Computer** to show the top-level folder, which includes local disks and mounted volumes.

- **Always open folders in a new window**

  Select this option to display folder contents in a separate window when a user opens a folder.

- **Always open windows in column view**

  Select this option to display folders in column view, which maintains a consistent view across windows.

- **Show warning before emptying the Trash**

  Select this option to display the normal warning when a user empties the Trash, or deselect it if you don't want users to see this message.

- **Always show file extensions**

  Select this option to show filename extensions (such as .txt or .jpg) that identify the file type; or deselect it to hide filename extensions.

  Once enabled, this group policy takes effect when users log out and back in.

- **Configure Finder views**

  Enable this group policy to control Finder views, for example the arrangement and appearance of items on a user's desktop, in Finder windows, and in the top-level folder of the computer.

The options in **Desktop View** allow you to adjust the size and arrangement of icons on the desktop.

Use **Icon Size** to adjust the icon size.

Use **Icon Arrangement** to specify how to arrange icons:

- To keep items aligned in rows and columns, select **Snap to grid**.
- To arrange items by criteria such as name or type (for example, all folders grouped together), select **Keep arranged by**.

Items in Finder windows are viewed in a list or as icons and you can control aspects of how these items look.

Default View settings control the overall appearance of all Finder windows. **Computer View** settings control the view for the top-level computer folder, showing hard disks and disk partitions, external hard drives, mounted volumes, and removable media (such as CDs or DVDs).

In **Icon View**, use **Icon Size** to adjust the size of icons.

Use **Icon Arrangement** to specify how to arrange icons:

- To keep items aligned in rows and columns, select **Snap to grid**.
- To arrange items by criteria such as name or type (for example, all folders grouped together), select **Keep arranged by**.

In **List View**, set the following:

- Select **relative dates** to show an item's creation or modification date relative to today, rather than as a fixed date; for example, Today or Yesterday, instead of 3/24/10.
- Select **Calculate folder sizes** to calculate the total size of each folder shown in a Finder window, which can take a lot of time depending on the size of the folder.

In **Icon Size**, select Select **small** or **big** for the size of icons in list view.

Once enabled, this group policy takes effect when users log out and back in.

Folder Redirection

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection** group policies to redirect specified folders from a network home directory to the local computer.

When you set up a network home directory, all home directory files are written to the network share. Some folders, such as /Library/Caches, get heavy I/O from Apple and third-party applications, which may cause performance issues. The folder redirection policies enable you to redirect specific folders, such as /Library/Caches, to the local computers, which can result in dramatic performance improvements.

Folder Redirection contains two folders with identical sets of four policies:

- **Folder redirection actions at login time** applies the specified policy when the user logs in. For example, at login delete a folder in the network home directory and create a symbolic link to it on the local computer.
- **Folder redirection actions at logout time** applies the specified policy when the user logs out. For example, at logout, delete the symbolic link on the local computer (created at login) and restore the original folder to the network home directory.

After enabling the policy, click **Add**, then enter the following:

- **Path** The path to the folder on the network share. You do not need to specify the actual network share location — you can simply use the tilde (~) for the user's home directory; for example, ~/Library/Caches specifies the /Library/Caches directory in the user's network home directory.
- **Link** The location to create or delete on the local computer. For example:

  /tmp/Library/Caches
- If you wish, you can use the syntax %@ to specify the logged in user's name. For example:

  /tmp/%@/Library/Caches

If cain is the logged in user, the folder that is created is:

`/tmp/cain/Library/Caches`

The Folder Redirection policies are listed here and explained below:

- Delete path
- Delete symbolic link and restore
- Delete and create symbolic link
- Rename and create symbolic link

Delete path

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection > Folder Redirect Actions at Login/Logout Time > Delete path

Description

Deletes the specified directory from the network home directory. For example, to delete the /Library/Caches file from each user's home directory, enter the following in the **Path** box:

~/Library/Caches

Typically, you enable this policy for the **login time** folder.

> **Note**: You are not required to enter anything in the **Link** box for this group policy, and in fact, anything you enter in this box will be ignored. All the policies in this folder are implemented with the same UI and the other policies require the Link box so it appears for this policy as well.

Once this group policy is enabled, it takes effect when users log in (enabled for **login time** folder) or log out (enabled for **logout time** folder).

Delete Symbolic Link and Restore

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection > Folder Redirect Actions at Login/Logout Time > Delete symbolic link and restore

Description

Deletes a previously defined symbolic link on the local computer and restores the specified directory to the network home directory. Typically, you use this policy with the Rename and create symbolic link policy. For example:

At login (using Rename and create symbolic link) you save ~/Library/Caches in the network home directory to a temporary folder and redirect it to a folder on the local computer, for example /tmp/user/Library/Caches. At logout, you can enable Delete symbolic link and restore to delete the symbolic link and restore the folder on the network home directory, by specifying the following:

- **Path**: ~/Library/Caches
- **Link**: /tmp/%@/Library/Caches

where: %@ specifies the logged in user's name on the local computer.

Once this group policy is enabled, it takes effect when users log in (enabled for **login time** folder) or log out (enabled for **logout time** folder).

Delete and Create Symbolic Link

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection > Folder Redirect Actions at Login/Logout Time > Delete and create symbolic link

Description

Deletes the specified directory from the network home directory and creates a symbolic link to it on the local computer.

For example, to delete the user's /Library/Caches policy from the network home directory and create a link to it on the local computer, specify the following after enabling the policy:

- **Path**: ~/Library/Caches
- **Link**: /tmp/%@/Library/Caches

where %@ specifies the logged in user's name on the local computer. For example, if cain is the logged in user, the cache files are written to:

/tmp/cain/Library/Caches

Once this group policy is enabled, it takes effect when users log in (enabled for **login time** folder) or log out (enabled for **logout time** folder).

Rename And Create Symbolic Link

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Folder Redirection > Folder Redirect Actions at Login/Logout Time > Rename and create symbolic link

Description

Renames the specified directory in the network home directory to a temporary folder and creates a symbolic link to it on the local computer.

For example, to rename the user's /Library/Caches policy on the network home directory and create a link to it on the local computer, specify the following after enabling the policy for the **login time** folder:

- **Path:** ~/Library/Caches
- **Link:** /tmp/%@/Library/Caches

where %@ specifies the logged in user's name on the local computer. For example, if cain is the logged in user, the cache files are written to:

/tmp/cain/Library/Caches

To restore the original /Library/Caches directory, use the Delete symbolic link and restore policy (enabled for the **logout time** folder).

Once this group policy is enabled, it takes effect when users log in (enabled for **login time** folder) or log out (enabled for **logout time** folder).

Import Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Import Settings

Description

Mac OS X uses plist files to store application and other preferences. Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Import Settings** group policies to import plist files to customize your preferences:

- **Import plist files.** This group policy allows you to import preferences from another computer to computers in your Delinea-managed domain. To do so you:
  - Copy the plist files you want to use to the system volume on the domain controller.
  - Use the Import plist files group policy to import the plist files to computers in the domain.

This group policy automatically processes plist files to extract MCX settings when the files are imported.

- **Import MCX setting plist files.** This group policy is similar to the **Import plist file** group policy, except that it does not process any data from the inputted plist files. This group policy copies the exact content (that is, the "raw" content) from the plist file and imports it to the Active Directory user record.

When you import the plist files, Delinea copies them to the appropriate directories on the local computers to implement the preferences that they control.

You can gather and copy plist files from multiple computers and paste them to the sysvol directory on the domain controller, but a more structured approach is to set up a preferences 'template' computer, that is, a computer that is set up with your desired preferences. Then you can copy the appropriate plist files to sysvol on the domain controller. Finally, you can use either of the group policies described here to import the plist files to Delinea-managed computers in the domain.

Mac OS X stores plist files in the /Library/Preferences directory and in the /Users/userName/Library/Preferences directory.

The following section shows specifics of using these group policies.

Import plist Files

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Import Settings > Import plist files

Description

Specify the names of plist files to import from the system volume (SYSVOL) — similar to importing plist files in Mac Workgroup Manager. By default, the system volume folder is at: \\domain\SYSVOL\domain\plist.

Before enabling this policy, you should copy all the plist files to import to the system volume (sysvol) on the domain controller.

To add a file, select **Enabled**, click **Add**, then type a filename.

The path you type in **plist file** is relative to \\domain\SYSVOL\domain\plist. For example, if the domain name is ajax.org and you enter a plist file named com.apple.MCX.plist, the file that gets imported is:

\\ajax.org\sysvol\ajax.org \com.apple.MCX.plist

You can specify additional relative directories in the path, if needed.

Once this group policy is enabled, it takes effect when users log out and log back in.

Import MCX Setting plist Files

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Import Settings > Import MCX setting plist files

Description

Enable this group policy to import raw MCX settings plist files from SYSVOL. By default the folder is \\<domain>\SYSVOL\<domain>\mcxplist, similar to importing plist files in Mac Workgroup Manager.

The plist file path that you specify is relative to this path:

\\<domain>\SYSVOL\<domain>\mcxplist

For example, if you specify this path:

com.apple.MCX.plist

the following plist file is imported:

\\<domain>\SYSVOL\<domain>\mcxplist\com.apple.MCX.plist

This group policy is similar to "Import plist files". However, instead of extracting MCX settings from the plist file like "Import plist files" does, this policy imports the entire plist file without processing it.

An example plist file format is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>mcx_application_data</key>
   <dict>
```

```
    <key>TARGET</key>
    <dict>
      <key>Forced</key>
      <array>
        <dict>
          Settings
        </dict>
      </array>
    </dict>
  </dict>
</dict>
</plist>
```

In this example, TARGET is the targeted MCX settings (such as com.apple.dock or com.apple.finder)

The recommended way to obtain the plist file with the correct format is by using the dscl command, and reading the MCX settings attribute of the user object that has the same MCX settings configured. Then copy the exact MCX settings and paste them into a plist file.

For example:

```
dscl /CentrifyDC read /Users/XXXX MCXSettings
```

where *XXXX* is an Active Directory user with the desired MCX settings.

Login Settings

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Login Settings

Description

Use the **User Configuration › Policies › Centrify Settings › Mac OS X Settings › Login Settings** group policy to specify frequently used items, such as applications, folders, or server connections to automatically open when a user logs in.

After enabling this policy, you can do the following:

- Use the **Add** button to specify the path to applications to open.
- In the **Network Home** area, use the **Add** button to specify URLs for servers to connect to; use the check box to specify whether to automatically connect the logged in user to the specified servers.
- Use the other check boxes to control whether users have the ability to add or remove login items.

The following table shows specifics of using this group policy.

> **Note**: Only the **Login items** area is visible when you first open the properties page for the group policy. Use the scroll bar to see the **Network share** area and other items that you can configure with this policy.

Enable Login Items

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Login Settings › Enable login items

Description

Specify the names of applications, folders, and server locations to open automatically when a user logs in. Select **Enable**, then do any or all of the following:

- **Login items**. To add an application to open automatically, click **Add**, then type the path to the application; for example:

```
/Applications/TextEdit.app
```

To initially hide the application, select **Hide**. The application will open, but its window and menu bar remain hidden until the user activates the application (for example, by clicking the application icon in the doc).

Click **OK** to save the item you entered. You can click **Add** as often as necessary to add multiple applications. You can also select an item in the window and click **Edit** to change it, or **Remove** to delete it.

- **Network share**. To add access to a network share, click **Add**, then type the URL in one of the following formats:

```
smb://server/share

smb://server/hidden$

smb://server/share/subdir

smb://user:password@server/share

smb://user:@server/share

afp://server/share

nfs://server/share

nfs://192.168.0.1/share
```

To automatically connect the user to the share with the user's login name and password, select **Authenticate selected share point with user's login name and password**.

> **Note**: If you uncheck this option, the share name must comply with RFC 1738 - Uniform Resource Locators (URL), which specifies that special characters need to be encoded, for example, by using `%20` instead of a space.

If the network share can be authenticated using Kerberos, this option can be ignored. If the network share cannot be authenticated using Kerberos, and this option is unchecked, then the user will be prompted for a username and password.

If a username is specified in the URL for the network share, then checking this option will still mount the share as the login user, while deselecting this option will mount the share as the user specified in the URL. For example, if network share is `smb://mount_user:password@server/share`, checking the option will mount the share as `login_user`, while deselecting the option will mount the share as `mount_user`.

Click **OK** to save the item you entered. You can click **Add** as often as necessary to add multiple shares. You can also select an item in the window and click **Edit** to change it, or **Remove** to delete it.

- Select **User may add and remove additional items** to allow users to add items to the list and remove items from the list.

  Deselect this box to prevent users from adding items or removing the items that you have specified. Note that they can remove login items that they specified on their own.

- Select **User may press Shift to keep items from opening** to allow user's to stop items from opening by holding down the Shift key during login until the Finder appears on the desktop.

  Deselect this option to prevent users from stopping applications from opening automatically.

Once enabled, this group policy takes effect when users log out and log back in.

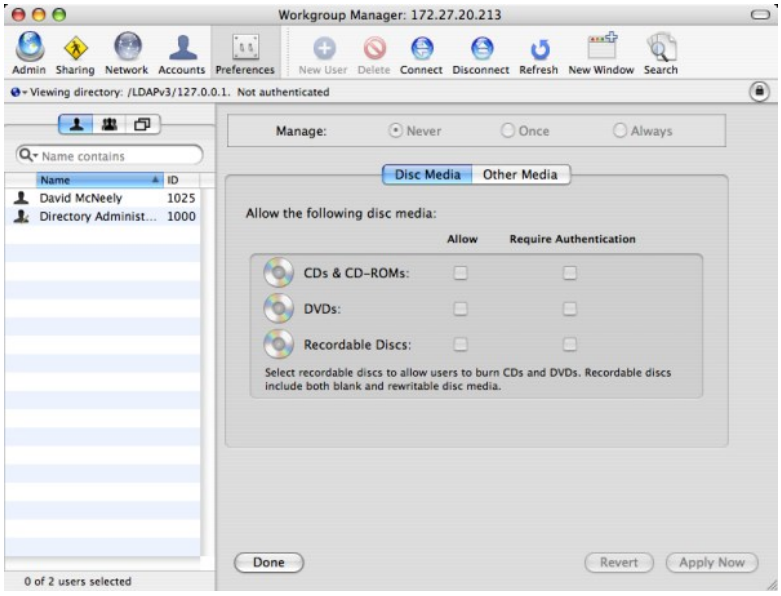
Media Access Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings** group policies to manage the access to discs and other media for Mac users. These group policies enable you to control access to specific types of media, such as CDs or DVDs, but you cannot restrict access to specific discs or to specific items, such as music or movies, on a disc type users are permitted to access. These settings correspond to the Media Access preferences you can manage using the Workgroup Manager. For example:



Permit/Prohibit Access: CDs and CD-ROMs

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Permit/prohibit access: CDs and CD-ROMs

Description

Control whether users can access data and applications on CDs and CD-ROMs. The valid options are:

- **allow** to allow access to CDs and CD-ROMs without authentication.
- **allow, require authentication** to require users to provide credentials for authentication before allowing them access to CDs and CD-ROMs.
- **deny** to prevent users from accessing any data or applications on CDs and CD-ROMs.

Once this group policy is enabled, it takes effect when users log out and log back in.

Permit/Prohibit Access: DVDs

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Permit/prohibit access: DVDs

Description

Control whether users can access data and applications on DVDs. The valid options are:

- **allow** to allow access to DVDs without authentication.
- **allow, require authentication** to require users to provide credentials for authentication before allowing them access to DVDs.
- **deny** to prevent users from accessing any data or applications on DVDs.

Once this group policy is enabled, it takes effect when users log out and log back in.

Permit/Prohibit Access: Recordable Discs

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Permit/prohibit access: Recordable Discs

Description

Control whether users can record or access data and applications on recordable discs. The valid options are:

- **allow** to allow access to recordable discs without authentication.
- **allow, require authentication** to require users to provide credentials for authentication before allowing them access to recordable discs.
- **deny** to prevent users from accessing any data or applications on recordable discs.

Allowing users access to recordable discs enables users to burn CDs and DVDs. Recordable discs can be blank or rewritable disc media.

Once this group policy is enabled, it takes effect when users log out and log back in.

Permit/Prohibit Access: Internal Discs

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Permit/prohibit access: Internal Discs

Description

Control whether users can access data and applications on internal discs. The valid options are:

- **allow** to allow read and write access to internal discs without authentication.
- **allow, read-only** to allow read-only access to the media.
- **allow, require authentication** to require users to provide credentials for authentication before allowing them access to the media.
- **allow, require authentication, read-only** to require users to provide credentials for authentication before allowing them access to internal discs, and grant **read-only access to the media** if authentication is successful.
- **deny** to prevent users from accessing any data or applications on internal discs.

Once this group policy is enabled, it takes effect when users log out and log back in.

Permit/Prohibit Access: External Discs

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Permit/prohibit access: External Discs

Description

Control whether users can access data and applications on external discs. External disks include floppy disks, FireWire drives, and all other external storage devices except CDs and DVDs. The valid options are:

- **allow** to allow read and write access to external discs without authentication.
- **allow, read-only to allow read-only access to** external discs.
- **allow, require authentication** to require users to provide credentials for authentication before allowing them access to external discs.
- **allow, require authentication, read-only** to require users to provide credentials for authentication before allowing them access to external discs, and grant **read-only access to the media** if authentication is successful.
- **deny** to prevent users from accessing any data or applications on external discs.

Once this group policy is enabled, it takes effect when users log out and log back in.

Eject All Removable Media at Logout

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Media Access Settings > Eject all removable media at logout

Description

Control whether removable media, such as CDs, DVDs, Zip disks, or FireWire drives, are automatically ejected when users log out. If you enable this group policy, CDs, DVDs, and other disk media are automatically ejected when users log out to ensure removable media is properly disconnected and put away when users end their sessions.

Once this group policy is enabled, it takes effect when users log out and log back in.

Mobility Settings

Use the **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Mobility Settings** group policies to control if macOS creates a Mobile Account for the Active Directory user when logging in.

Configure Mobile Account Creation

Path

Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Mobility Settings > Configure mobile account creation

Description

Configure whether mobile accounts are created when users log in.

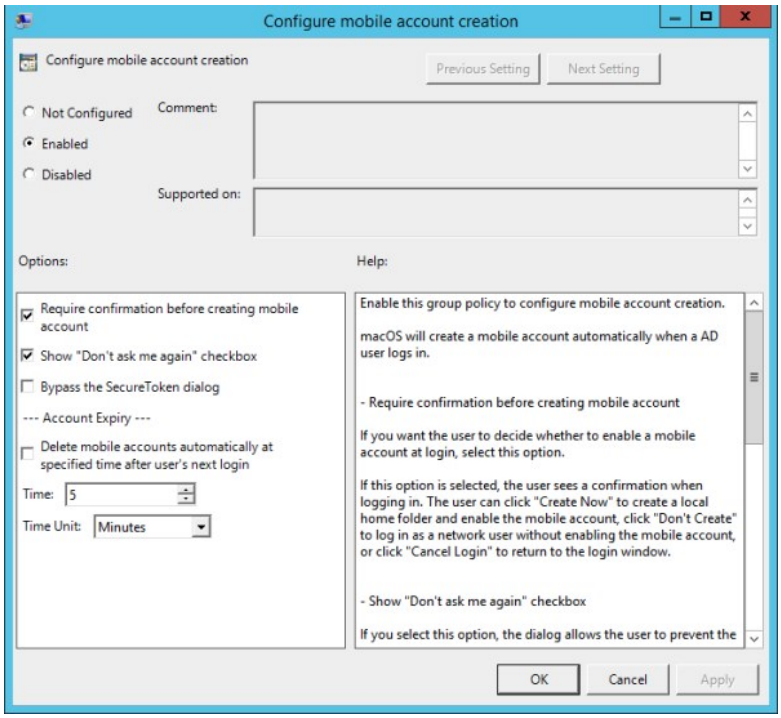
Check **Require confirmation before creating mobile account** to allow users to decide whether to enable a mobile account at login. Users see a confirmation dialog when logging in and can click one of the following:

- "Create Now" to create a local home folder and enable the mobile account.
- "Don't Create" to log in as a network user without enabling the mobile account.
- "Cancel Login" to return to the login window.

Select **Show "Don't ask me again" checkbox** to provide a check box that allows users to prevent display of the mobile account creation dialog on that computer in the future. Users who select "Don't ask me again" and click "Don't Create", are not asked to create a mobile account on that computer (unless they hold down the Option key during login to redisplay the dialog).

Select **Bypass the SecureToken dialog** so that the system will bypass the secure token authorization dialog. The Active Directory user can continue to create a mobile account. However, if this volume is encrypted, the Active Directory mobile user may not be able to log in with this account when the computer starts. This dialog only affects APFS volumes.

Select **Delete mobile accounts automatically at specified time after user's next login** so that MacOS will delete the mobile account and its local home folder automatically after a period of inactivity.



Printing settings

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Printing Settings › Specify printer list (with model)

Description

Use the **User Configuration › Policies › Centrify Settings › Mac OS X Settings › Printing Settings › Specify printer list (with model)** group policy to specify a list of printers for a user.

The printers that are available to a user are a combination of those specified in this policy and those added through System Preferences on the local computer. Note that this policy allows an administrator to control whether the user can add or see printers on the local computer, or is only allowed to use the managed printers specified by this policy.

Specify a managed list of network printers that are available to a user on this computer. Printers specified by this policy use a generic PostScript driver.

To add a printer, click **Add** and enter the following information:

- **Name**: A name of your choosing for the printer.
- **DeviceURI**: The device Uniform Resource Identifier, which specifies the device that is assigned to the printer (see **Specifying the Device URI**, below); for example:
  - socket://192.168.0.20:9100 (which identifies the protocol, IP address, and port number)
  - cdcsmb://dc1.acme.com/HPLaserJet2 (which identifies a Windows printer added using the Delinea protocol and identified by hostname.)
- (Optional) **Model**: The printer driver for the printer model (see **Specifying the Model (printer driver)**, below); for example: HP Photosmart C6100 series. Fax

You can use the following options to control access to the printers on the local computer:

- **Allow user to modify the printer list**: Check this option to allow local users to make changes in System Preferences to the printers that have been added by this policy, including deleting them.

  Deselect this option to prevent local users from modifying the printers added by this policy.

- **Allow printers that connect directly to user's computer**: Check this option to allow users to add their own local printers.

  Deselect this option to prevent users from adding local printers.

- **Require an administrator password**: Check this option to require an administrator's password when adding local printers.

- **Only show managed printers**: Check this option to allow local users to use only the managed printers specified by this option.

  Printers added locally, for example, through System Preferences, will not be visible.

  Deselect this option to allow local users to use printers added locally, as well as the managed printers added by this policy.

Printers added through this group policy appear after the next group policy refresh interval.

Specifying the Device URI

When you add a printer through the **Specify printer list** group policy, or locally by using the **Print & Scan, Add Printer** advanced options, the printer is implemented through the Common UNIX Printing System (CUPS), which was developed by Apple for Mac OS X and other UNIX-like operating systems.

The CUPS system supports the following device Uniform Resource Identifier (URI) protocols that you can use to specify the printers to add.

AppSocket or Jetdirect Protocol

The AppSocket, or JetDirect, protocol normally prints over port 9100 and uses the socket URI scheme:

socket://ip-address-or-hostname

socket://ip-address-or-hostname:port-number

Internet Printing Protocol (IPP)

CUPS supports IPP natively. IPP printing normally happens over port 631 and uses the http and ipp URI schemes:

ipp://ip-address-or-hostname/resource

ipp://ip-address-or-hostname:port-number/resource

http://ip-address-or-hostname:port-number/resource

Line Printer Daemon Protocol (LPD)

LPD is the original network printing protocol and is supported by many network printers. LPD printing normally happens over port 515 and uses the lpd URI scheme:

lpd://ip-address-or-hostname/queue

lpd://username@ip-address-or-hostname/queue

Windows Printer via Delinea

When Mac users print on a Windows network printer, they must authenticate separately. Specifying a Windows printer via Delinea allows users to access the printer without providing credentials as they have already been authenticated through Active Directory.

Delinea printing normally happens over port 445 and uses the cdcsmb URI scheme

cdcsmb://server_fqdn/printersharename

Windows

Windows printing normally happens over port 445 and uses the smb URI scheme:

> **Note**: You can use the Delinea protocol (cdcsmb), if you want to use Windows network printers without providing credentials each time.

smb://workgroup/server/printersharename

smb://ip-address-or-hostname/printersharename

smb://username:password@workgroup/ip-address-or-hostname/printersharename

smb://username:password@ip-address-or-hostname/printersharename

Specifying the Model (printer driver)

Model specifies the model name of the added printer and is used to determine which device driver to associate with the printer. Be certain to specify model correctly, otherwise, if model is not specified, or does not match a driver installed on the client Mac OX X computer, Generic PostScript driver will be selected for the printer, which may result in fewer printing options.

To find the correct model name, take one of these two approaches:

Use Printers & Scanners to identify the model:

1. On a Mac OX X computer, open **System Preferences > Printers & Scanners**.
2. Click **Add (+)** to add a printer.
3. When you select a printer, the correct model name appears on the "Use" drop down menu.

Use lpinfo to identify the model

1. On a Mac OX X computer, open the Terminal application.
2. Run the following command to obtain the list of all the models available:

   "lpinfo -m" command

   In the output from lpinfo, the correct model string appears right after *.ppd.gz. For example:

   Library/Printers/PPDs/Contents/Resources/HP Photosmart C6100 Series Fax.ppd.gz HP Photosmart C6100 series. Fax

   The model string is:

   HP Photosmart C6100 series. Fax
3. Type this in the group policy's **Model** field.

Scripts (Login/Logout)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout)

Description

Use the **User Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout)** group policies to deploy login and logout scripts that run when an Active Directory user logs on or logs out. When you use these group policies, the login and logout scripts are stored in the Active Directory domain's system volume (sysvol) and transferred to the Mac computer when the group policies are applied. Login and logout scripts are useful for performing common tasks such as mounting and un-mounting shares.

> **Note**: When these group policies are enabled, the first login by an AD user will restart the login script and return the user to the login window. Subsequent logins by this user or a different user occur normally and the changes generated by the script happen immediately.

Specify Login Script (Deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout) > Specify login script

Description

Specify the name of a login script to execute when users log on. You can specify only one file as the login script.

Before enabling this policy, you should create the login script and copy it to the system volume (sysvol) on the domain controller. By default, the login script is stored in the system volume (SYSVOL) on the domain controller in the directory:

```
\\domain\SYSVOL\domain\Scripts\scriptname
```

The script path you type in **Login script** is relative to \\domain\SYSVOL\domain\scripts\. For example, if the domain name is ajax.org and you enter a script name of mlogin.sh, the script that gets executed on the domain controller is:

```
\\ajax.org\SYSVOL\ajax.org\Scripts\mlogin.sh
```

You can specify additional relative directories in the path, if needed.

> **Note**: Be certain authenticated users have permission to read this file so the script can run when they log in.

By default, the script runs with the Active Directory user's permissions. If the script contains commands that require root permission to run, select **Run with root user privileges**.

Once this group policy is enabled, it takes effect when users log out and log back in.

> **Note**: The first AD user to log in is taken back to the login screen. Subsequent logins by this user or a different user occur normally and changes generated by the script happen immediately.

Specify Logout Script

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout) > Specify logout script

Description

Specify the name of a logout script to execute when users log out. You can specify only one file as the logout script.

Before enabling this policy, you should create the logout script and copy it to the system volume (SYSVOL) on the domain controller. By default, the logout script is stored in the system volume (SYSVOL) on the domain controller in the following directory:

```
\\domain\SYSVOL\domain\Scripts\scriptname
```

The script path you type in **Logout script** is relative to: \domain\SYSVOL\domain\scripts\.

For example, if the domain name is ajax.org and you enter a script name of mlogout.sh, the script that gets executed on the domain controller is:

```
\\ajax.org\SYSVOL\ajax.org\Scripts\mlogout.sh
```

> **Note**: Be certain authenticated users have permission to read this file so the script can run when they log out.

By default, the script runs with the Active Directory user's permissions. If the script contains commands that require root permission to run, select **Run with root user privileges**.

Once this group policy is enabled, it takes effect when users log out and log back in.

Specify Multiple Login Scripts

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Scripts (Login/Logout) > Specify multiple login scripts

Description

Specify the names of one or more login scripts to execute when a user logs on. The scripts you specify run simultaneously in no particular order.

This policy is also available as a computer policy. If you specify scripts using both the computer and user policies, the computer scripts are executed first.

Before enabling this policy, you should create the scripts and copy them to the system volume (sysvol) on the domain controller. By default, the login scripts are stored in the system volume (SYSVOL) on the domain controller in the directory:

```
\\domain\SYSVOL\domain\Scripts
\scriptname1
\scriptname
...
```

After enabling this policy, click **Add** and enter the following information:

- ___Script:___The name of the script and an optional path, which are relative to \\domain\SYSVOL\domain\scripts\.

For example, if the domain name is ajax.org and you enter a script name of mlogin.sh, the script that gets executed on the domain controller is:

`\ajax.org\SYSVOL\ajax.org\Scripts\mlogin.sh``

You can specify additional relative directories in the path, if needed; for example, if you type submlogin.sh, the file that gets executed is:

```
\\ajax.org\SYSVOL\ajax.org\Scripts\sub\mlogin.sh
```

- **Parameters**: An optional set of arguments to pass to the script.

These arguments are interpreted the same way as in a UNIX shell; that is, space is a delimiter, and backslash is an escape character. You can also use $USER to represent the current user's name. For example:

```
arg1 arg2 arg3
arg1 'a r g 2' arg3
arg' $USER.
```

> **Note**: Be certain authenticated users have permission to read these files so the scripts can run when they log in.

Once this group policy is enabled, it takes effect when users log out and log back in.

Security & Privacy Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy

Description

Use the Security & Privacy group policies to control user security and privacy settings.

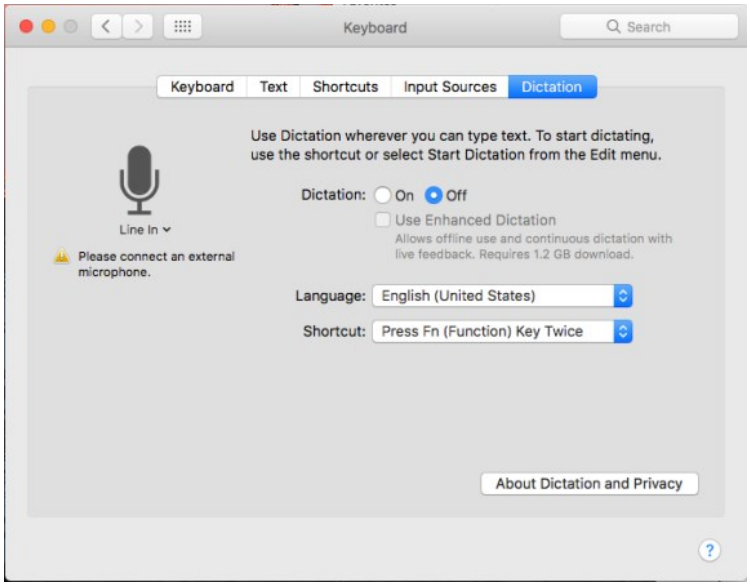
Disable Dictation

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › Disable Dictation

Description

Enable this policy to turn off Dictation in the System Preferences › Keyboard pane.



Once enabled, this group policy takes effect dynamically at the next group policy refresh interval.

Require a Password to Wake this Computer from Sleep or Screen Saver (Deprecated)

> **Note**: This group policy is deprecated, and will not work anymore. Please use the new same name group policy under "Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › Require a password to wake this computer from sleep or screen saver" instead.

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › Require a password to wake this computer from sleep or screen saver

Description

Lock the computer screen when the computer goes into sleep or screen saver mode and requires users to enter a user name and password to unlock the screen.

Enabling this group policy is the same as clicking the **Require a password to wake this computer from sleep or screen saver** option in the Security system preference.

After this group policy is enabled, it takes effect when the computer is rebooted.

Prohibit Authentication with Expired Password

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › Prohibit authentication with expired password

Description

Prohibit a user from unlocking the screen if a password change is required while the screen is locked. If a user logs in with a password that must be changed, and the computer goes into sleep or screen saver mode before the user updates the password, the user is locked out. Disabling this policy allows a user to

specify the old password to remove the screen lock.

Keychain Policies

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Keychain Policies

Description

On OS X 10.11, you can create a keychain protected by a smart card token or a password. Once the Enable smart card protected keychain group policy takes effect, the token-protected keychain can only be unlocked with a PIN when the associated smart card is present. This group policy can be configured to allow users who lose or forget their smart card to continue to log in with a password. In this case, a new password-protected keychain is created to ensure users can continue to log in to their account; however, keychain items are not transferred from the token-protected keychain to the password-protected keychain.

This feature is not supported on OS X 10.10 and earlier.

Enable Protected Keychain

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Keychain Policies > Enable protected keychain

Description

Create a new keychain protected by either an asymmetric token stored on a smart card or by a password, depending on the log in type. Enabling this policy requires users to have the smart card present to unlock the token-protected keychain.

When the smart card is renewed it will no longer unlock the protected keychain. There is no way to export a token-protected keychain; you will have to recreate the keychain items in the new token-protected keychain. In addition, if a smart card is lost, there is no way to recover items from the token-protected keychain.

The **Set as user default keychain** option is selected by default. This option is required to be able to log in with a password after this group policy takes effect. With this option set, the default keychain will be switched based on the login type (smart card login or password login). Deselect this option to leave the existing login keychain as the default keychain.

The **Delete the Password protected 'Login' Keychain after login** option is deselected by default. Select this option to delete the existing password-protected 'Login' Keychain after logging in with a smart card, leaving no keychains that can be unlocked without a smart card. This option is required to be able to log in with a password after this group policy takes effect without seeing keychain errors.

> **Note**: This feature is not supported on OS X 10.10 and earlier.

Lock Protected Keychain After Number of Minutes of Inactivity

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Keychain Policies > Lock protected keychain after number of minutes of inactivity

Description

Lock the protected keychain after a period of inactivity that you specify in minutes.

This policy only works if you have enabled the Enable protected keychain group policy.

This policy takes effect at the next user login using smart card authentication.

Lock Protected Keychain When Sleeping

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > Keychain Policies > Lock protected keychain when sleeping

Description

Lock the protected keychain when the machine sleeps.

This policy only works if you have enabled the Enable protected keychain group policy.

This policy takes effect at the next user login using smart card authentication.

Allow All Applications to Access The Auto-Enrollment Private Key(S)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Allow all applications to access the auto-enrollment private key(s)

Description

Enabling this policy allows all applications to access the auto-enrollment private key(s) in the System keychain.

See Configuring Auto-Enrollment for more information about auto-enrollment keys.

> **Note**: This setting only applies to new auto-enrollment private key(s); it will not update already imported auto-enrollment private key(s) that are in the System keychain.

Allow Specific Applications to Access the Auto-Enrollment Private Key(S)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy > Public Key Policies > Allow specific applications to access the auto-enrollment private key(s)

Description

Enabling this policy allows specified applications to access the auto-enrollment private key(s) in System keychain.

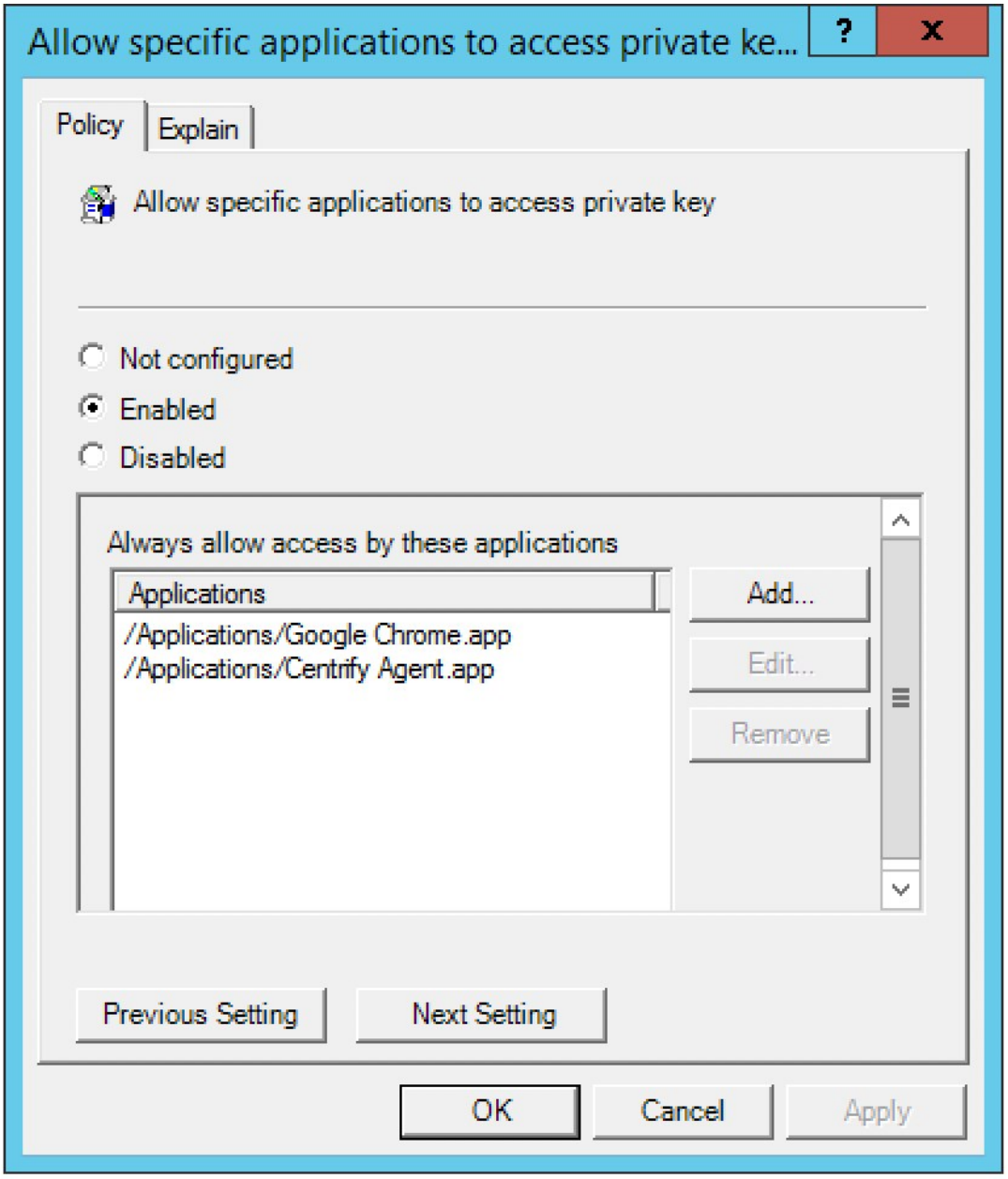
After you enable this policy, click **Add** to enter the path to the application you want to allow access to the auto-enrollment private key, then click **OK**. You can click **Add** again to add additional applications.

For example, to give Google Chrome and Delinea Agent access to the auto-enrollment private key, enter the application path for Google Chrome:

```
/Applications/Google Chrome.app
```

Click **OK**. Then click **Add** and enter the application path for Delinea Agent:

```
/Applications/Centrify Agent.app
```



After this group policy is enabled, the list of applications specified in the group policy are added to the access control list of the auto-enrollment private key in system keychain.

See Configuring Auto-Enrollment for more information about auto-enrollment keys.

> **Note**: This setting only applies to a new auto-enrollment private key. It does not change auto-enrolled private keys that are already in the keychain.

If the group policy above, **Allow All Applications to Access the Auto-enrollment Private Key(s)** is enabled, this group policy will be ignored.

Do Not Allow the Private Key(S) To Be Extractable

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy › Public Key Policies › Do not allow private key(s) to be extractable

Description

Enabling this policy prevents exporting the auto-enrollment private key(s).

> **Note**: This setting only applies to new auto-enrollment private key(s). It does not change auto-enrolled private keys that are already in the keychain.

System Preference Settings

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › System Preference Settings

Description

Use the **User Configuration › Policies › Centrify Settings › Mac OS X Settings › System Preference Settings** group policies to specify which preferences are enabled for use in System Preferences for Mac OS X users. Enabling a preference for use does not enable non-admin users to modify that preference. For example, some preferences, such as Startup Disk preferences, require an administrator name and password before a user can modify its settings. Displaying a preference does enable a user to view the preference's current settings.

By default, no system preference panes are displayed unless explicitly enabled. The group policies in this category correspond to System Preferences you can select for display in the Workgroup Manager. For example:

The user interface for System Preferences Settings differs significantly between different versions of Mac OS X. Therefore, there are separate System Preferences policies for each supported version of Mac OS X. In addition, to support existing installations that configured group policies by using a previous centrifydc_mac_settings template, the Centrify group policies provide a set of legacy preferences settings.

The **Use Version Specific Settings** group policy below determines whether to use legacy settings or platform-specific system preferences settings. By default (if you do not configure or disable this policy) legacy settings are used.

If you enable this policy, you can then enable platform-specific system preferences settings for each platform in your environment; see the following sections for information on each set of policies:

Use Version Specific Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Use version specific settings

Description

Enable the use of version-specific System Preferences settings.

If you enable this policy, you can then set platform-specific preferences settings for each platform in your environment. For example, if you have only 10.10 computers, you can enable this policy and then use Mac OS X 10.10 settings. If you have 10.9 and 10.10 computers, enable this policy, and then configure the version-specific policies as appropriate:

- Mac OS X 10.9 Settings
- Mac OS X 10.10 Settings

When a computer joins the domain, Delinea Management Services determines the OS version and applies the appropriate Preferences settings.

If this policy is disabled or not configured, Legacy Settings are used instead of version-specific settings. Likewise, Delinea versions prior to 4.4.2 always use Legacy Settings and ignore this policy setting.

If you configured System Preferences settings with a version of the product prior to 4.4.2, these settings are saved to Legacy Settings when you upgrade to the current version. You can keep or edit these settings as you wish.

> **Note**: The Legacy Settings may not match exactly the settings for each OS version; for example, some settings may be missing while others may be redundant for a particular OS version.

Legacy Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Legacy Settings

Description

When you upgrade from a version of Delinea prior to 4.4.2, your System Preferences settings are saved to Legacy Settings. You can keep or edit the individual legacy system preferences group policy settings as you wish.

> **Note**: The legacy settings may not match exactly the settings for each OS version; for example, some settings may be missing while others may be redundant for a particular OS version.

Showing items in the Personal pane of System Preferences	Select the items to display in the Personal pane of System Preferences.
Showing items in the Hardware System pane of Preferences	Select the items to display in the Hardware pane of System Preferences.
Showing items in the Internet & Network pane of System Preferences	Select the items to display in the Internet & Network pane of System Preferences.
Showing items in the System pane of System Preferences	Select the items to display in the System pane of System Preferences.
Showing items in the Other pane of System Preferences	Select the items to display in the Other pane of System Preferences.

Limit items usage in System Preferences (deprecated)	Limit the usage of items in System Preferences. You must enable this group policy for any of the other group policy settings to take effect. Once this group policy is enabled, it takes effect when users log out and log back in.

Showing items in the Personal pane of System Preferences

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Enable System Preferences Pane: Personal

Description

Use the group policies in this category to choose which items to display in the Personal pane of System Preferences.

Enable Appearance

Enable usage of Appearance preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Dashboard & Expose

Enable usage of Dashboard & ExposE preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Desktop & Screen Saver

Enable usage of Desktop & Screen Saver preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Dock

Enable usage of Dock preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable International (Language & Text)

Enable usage of International preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Security

Enable usage of Security preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Spotlight

Enable usage of Spotlight preferences in the Personal pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Showing items in the Hardware System pane of Preferences

Use the group policies in this category to display items in the Hardware pane of System Preferences.

Enable Bluetooth

Enable usage of Bluetooth preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable CDs & DVDs

Enable usage of CDs & DVDs preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Displays

Enable usage of Displays preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Energy Saver

Enable usage of Energy Saver preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Ink

Enable usage of Ink preferences in the Hardware pane of System Preferences.

> **Note**: Ink preferences are only shown if a graphics tablet is connected to the Mac computer.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Keyboard & Mouse (Keyboard)

Enable usage of Keyboard & Mouse preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Mouse

Enable usage of Mouse preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Print & FAX

Enable usage of Print & FAX preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Sound

Enable usage of Sound preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Trackpad

Enable usage of Trackpad preferences in the Hardware pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Showing Items in the Internet & Network Pane of System Preferences

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Enable System Preferences Pane: Internet & Network

Description

Use the group policies in this category to display items in the Internet & Network pane of System Preferences.

Enable .Mac (MobileMe)

Enable usage of .Mac preferences in the Internet & Network pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Fibre Channel

Enable usage of Fibre Channel preferences in the Internet & Network pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Network

Enable usage of Network preferences in the Internet & Network pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable QuickTime

Enable usage of QuickTime preferences in the Internet & Network pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Sharing

Enable usage of Sharing preferences in the Internet & Network pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Showing items in the System pane of System Preferences

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Enable System Preferences Pane: System

Description

Use the group policies in this category to display items in the System pane of System Preferences.

Enable Accounts

Enable usage of Accounts preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Classic

Enable usage of Classic preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Date & Time

Enable usage of Date & Time preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Parental Controls

Enable usage of Parental Controls preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Software Update

Enable usage of Software Update preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Speech

Enable usage of Speech preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Startup Disk

Enable usage of Startup Disk preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Time Machine

Enable usage of Time Machine preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Enable Universal Access

Enable usage of Universal Access preferences in the System pane of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

Showing Items in the Other Pane of System Preferences

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Enable System Preferences Pane: Other

Description

Use the group policies in this category to display the items you specify in the Other pane of System Preferences.

Other Preferences Panes

Enable usage of additional preferences panes of System Preferences.

Once this group policy is enabled, it takes effect when users log out and log back in.

System Preferences Mac OS X 10.5 Settings (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.5 Settings

Description

If your environment does not contain Mac OS X 10.5 computers, you can ignore the group policies in this folder.

System Preferences Mac OS X 10.6 Settings (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.6 Settings

Description

If your environment does not contain Mac OS X 10.6 computers, you can ignore the group policies in this folder.

System Preferences Mac OS X 10.7 Settings (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.7 Settings

Description

The Mac OS X 10.7 Settings allow you to configure system preferences policies that apply specifically to Mac OS X 10.7 computers. Because the user interface varies between different versions of Mac OS X, separate policies are provided for each version. See **Legacy Settings**, above, for older versions of Mac OS X.

If your environment does not contain Mac OS X 10.7 computers, you can ignore these settings.

Limit Items Usage in System Preferences (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.7 Settings > Limit items usage in System Preferences

Description

Limit the usage of items in the System Preferences panel.

Once enabled, this group policy takes effect when users log out and log back in.

Enable System Preferences Panes 10.7 (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.7 Settings > Enable System Preferences Panes

Description

Use **Enable built-in System Preferences Panes**, below, to select the items to add to the standard System Preferences panes.

Use **Enable other System Preferences Panes**, below, to add preferences for third-party applications to the Other pane of the System Preferences.

Enable Built-in System Preferences Panes (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.7 Settings >

Enable System Preferences Panes > Enable built-in System Preferences Panes

Description

Select items to add to the System Preferences panel.

This policy is only effective if the **Limit items usage in System Preferences** group policy, above, is enabled. Otherwise this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

Enable Other System Preferences Panes (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.7 Settings > Enable System Preferences Panes > Enable other System Preferences panes

Description

Define a list of additional items to add to the Other pane of the System Preferences panel.

Preference pane applications are actually collections of files inside a directory (called bundles). Inside the Contents directory of every preference pane application is the info.plist file, and inside that file is the CFBundleIdentifier key that identifies the preference pane application. You need to use the value for this key when adding a preference pane application.

Generally, installed third party preference panes can be found in /System/Library/PreferencePanes, /Library/PreferencePanes or ~/Library/PreferencePanes.

You can find the CFBundleIdentifier key by using the defaults command. For example, to find the value for the QuickTime pane, use the following command in a terminal window:

```
defaults read /System/Library/PreferencePanes/QuickTime.prefPane   /Contents/info CFBundleIdentifier
```

which returns:

```
com.apple.preference.quicktime
```

To display the QuickTime icon in the **Other** pane of the System Preferences Panel, enable this policy, then click **Add** and enter com.apple.preference.quicktime.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, below, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

System Preferences Mac OS X 10.8 Settings (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.8 Settings

Description

The Mac OS X 10.8 Settings allow you to configure system preferences policies that apply specifically to Mac OS X 10.8 computers. Because the user interface varies between different versions of Mac OS X, separate policies are provided for each version. See Legacy Settings for older versions of Mac OS X.

If your environment does not contain Mac OS X 10.8 computers, you can ignore these settings.

Limit Items Usage in System Preferences (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.8 Settings > Limit items usage in System Preferences

Description

Limit the usage of items in the System Preferences panel.

Once enabled, this group policy takes effect when users log out and log back in.

Enable System Preferences Panes 10.8 (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.8 Settings > Enable System Preferences Panes

Description

Use **Enable Built-in System Preferences Panes**, below, to select the items to add to the standard System Preferences panes.

Use **Enable Other System Preferences Panes**, below, to add preferences for third-party applications to the Other pane of the System Preferences.

Enable Built-in System Preferences Panes (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.8 Settings > Enable System Preferences Panes > Enable built-in System Preferences panes

Description

Select items to add to the System Preferences panel.

This policy is effective only if the Limit Items Usage on System Preferences group policy is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

Enable Other System Preferences Panes (deprecated)

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.8 Settings > Enable System Preferences Panes > Enable other System Preferences panes

Description

Define a list of additional items to add to the Other pane of the System Preferences panel.

Preference pane applications are actually collections of files inside a directory (called bundles). Inside the `Contents` directory of every preference pane application is the `info.plist` file, and inside that file is the `CFBundleIdentifier` key that identifies the preference pane application. You need to use the value for this key when adding a preference pane application.

Generally, installed third party preference panes can be found in `/System/Library/PreferencePanes`, `/Library/PreferencePanes`, or `~/Library/PreferencePanes`.

You can find the `CFBundleIdentifier` key by using the `defaults` command. For example, to find the value for the QuickTime pane, use the following command in a terminal window:

```
defaults read /System/Library/PreferencePanes/QuickTime.prefPane   /Contents/info CFBundleIdentifier
```

which returns:

```
com.apple.preference.quicktime
```

To display the QuickTime icon in the **Other** pane of the System Preferences Panel, enable this policy, then click **Add** and enter `com.apple.preference.quicktime`.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, below, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

System Preferences Mac OS X 10.9 Settings (deprecated)

Path

Description

The Mac OS X 10.9 Settings allow you to configure system preferences policies that apply specifically to Mac OS X 10.9 computers. Because the user interface varies between different versions of Mac OS X, separate policies are provided for each version. See **Legacy Settings**, above, for older versions of Mac OS X.

If your environment does not contain Mac OS X 10.9 computers, you can ignore these settings.

Limit Items Usage in System Preferences (deprecated)

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › System Preference Settings › Mac OS X 10.9 Settings › Limit items usage in System Preferences

Description

Limit the usage of items in the System Preferences panel.

Once enabled, this group policy takes effect when users log out and log back in.

Enable Built-in System Preferences Panes (deprecated)

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › System Preference Settings › Mac OS X 10.9 Settings › Enable System Preferences Panes › Enable built-in System Preferences panes

Description

Select items to add to the System Preferences panel.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, below, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

Enable Other System Preferences Panes (deprecated)

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › System Preference Settings › Mac OS X 10.9 Settings › Enable System Preferences Panes › Enable other System Preferences panes

Description

Define a list of additional items to add to the Other pane of the System Preferences panel.

Preference pane applications are actually collections of files inside a directory (called bundles). Inside the `Contents` directory of every preference pane application is the `info.plist` file, and inside that file is the `CFBundleIdentifier` key that identifies the preference pane application. You need to use the value for this key when adding a preference pane application.

Generally, installed third party preference panes can be found in `/System/Library/PreferencePanes`, `/Library/PreferencePanes` or `~/Library/PreferencePanes`.

You can find the `CFBundleIdentifier` key by using the `defaults` command. For example, to find the value for the QuickTime pane, use the following command in a terminal window:

```
defaults read /System/Library/PreferencePanes/QuickTime.prefPane  /Contents/info CFBundleIdentifier
```

which returns:

```
com.apple.preference.quicktime
```

To display the QuickTime icon in the **Other** pane of the System Preferences Panel, enable this policy, then click **Add** and enter

com.apple.preference.quicktime.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, below, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

System Preferences Mac OS X 10.10 or Above Settings

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.10 Settings

Description

The Mac OS X 10. 10 or above Settings allow you to configure system preferences policies that apply specifically to Mac OS X 10.10 and above computers. Because the user interface varies between different versions of Mac OS X, separate policies are provided for each version. See **Legacy Settings**, above, for other versions of Mac OS X.

If your environment does not contain Mac OS X 10.10 or above computers, you can ignore these settings.

Limit Items Usage on System Preferences

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.10 Settings > Limit items usage on System Preferences

Description

Limit the usage of items in the System Preferences panel.

Once enabled, this group policy takes effect when users log out and log back in.

Enable System Preferences Panes 10.10

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.10 Settings > Enable System Preferences Panes

Description

Use **Enable other System Preferences Panes**, below, to add preferences for third-party applications to the Other pane of the System Preferences.

Enable Built-in System Preferences Panes

Path

User Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy Settings > System Preference Settings > Mac OS X 10.10 Settings > Enable System Preferences Panes > Enable built-in System Preferences panes

Description

Enable this group policy to enable the built-in System Preferences panes.

Enable or disable usage of items in the built-in System Preferences panes by checking or unchecking boxes corresponding to the items.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, above, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

Enable Other System Preferences Panes

Path

User Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy Settings › System Preference Settings › Mac OS X 10.10 Settings › Enable System Preferences Panes › Enable other System Preferences panes

Description

Define a list of additional items to add to the Other pane of the System Preferences panel.

Preference pane applications are actually collections of files inside a directory (called bundles). Inside the Contents directory of every preference pane application is the info.plist file, and inside that file is the CFBundleIdentifier key that identifies the preference pane application. You need to use the value for this key when adding a preference pane application.

Generally, installed third party preference panes can be found in /System/Library/PreferencePanes, /Library/PreferencePanes or ~/Library/PreferencePanes.

You can find the CFBundleIdentifier key by using the defaults command. For example, to find the value for the QuickTime pane, use the following command in a terminal window:

```
defaults read /System/Library/PreferencePanes/QuickTime.prefPane   /Contents/info CFBundleIdentifier
```

which returns:

```
com.apple.preference.quicktime
```

To display the QuickTime icon in the **Other** pane of the System Preferences Panel, enable this policy, then click **Add** and enter com.apple.preference.quicktime.

This policy is effective only if the **Limit Items Usage on System Preferences** group policy, above, is enabled. Otherwise, this group policy is ignored.

Once enabled, this group policy takes effect when users log out and log back in.

This section explains how to set up smart card login for a Mac computer:

Understanding Smart Card Login

Smart cards provide an enhanced level of security authentication for logging into an Active Directory domain. To configure a smart card for use on a Mac computer that is running the DirectControl agent, requires that you have already set up a smart card for use in a Windows domain. You do not need to add any smart card infrastructure to the Mac computer, other than a smart card reader and a provisioned smart card.

If you have set up smart card login for Windows clients in a domain, you can use Access Manager to configure smart card login for Mac clients joined to the same domain. If you have provisioned a smart card for use on a Windows computer once you configure smart card support for a Mac computer, you can use the same smart card to log in to a Mac computer.

Supported Smart Card Types

Delinea Smart Card Support for MacOS is based on the MacOS modern native framework, CryptoTokenKit. TokenD is no longer supported.

For macOS 10.15 and later, Delinea supports personal identity verification (PIV) smart cards, USB CCID class-compliant readers, and hard tokens that support the PIV standard.

Configuring Smart Card Login

Delinea provides group policies, configuration options, and account options to perform the smart card configuration tasks desribed below.

> **Note**: Before configuring smart card login, refer to the next section, **Verifying Prerequisites for Configuring Smart Card Login** to ensure your environment meets all the prerequisites.

Verifying Prerequisites for Configuring Smart Card Login

- Make sure that your smart card is supported by MacOS.

  MacOS 10.15 and later supports personal identity verification (PIV) smart cards, USB CCID class-compliant readers, and hard tokens that support the PIV standard.

  - Provision a smart card with an NT principal name and PIN.
  - Verify that the Active Directory user's UPN matches the UPN on the smart card.
  - Make sure that there are at least two certificates in your smart card; these two certificates are for two different purposes: "Signature and smartcard logon" and "Encryption." MacOS will use the certificate which purpose is "Signature and smartcard logon" to logon, and use the certificate which purpose is "Encryption" to encrypt and decrypt the user's Keychain automatically. If there is no certificate which is for "Encryption", the user will need to input the Keychain password every time when that they log in.

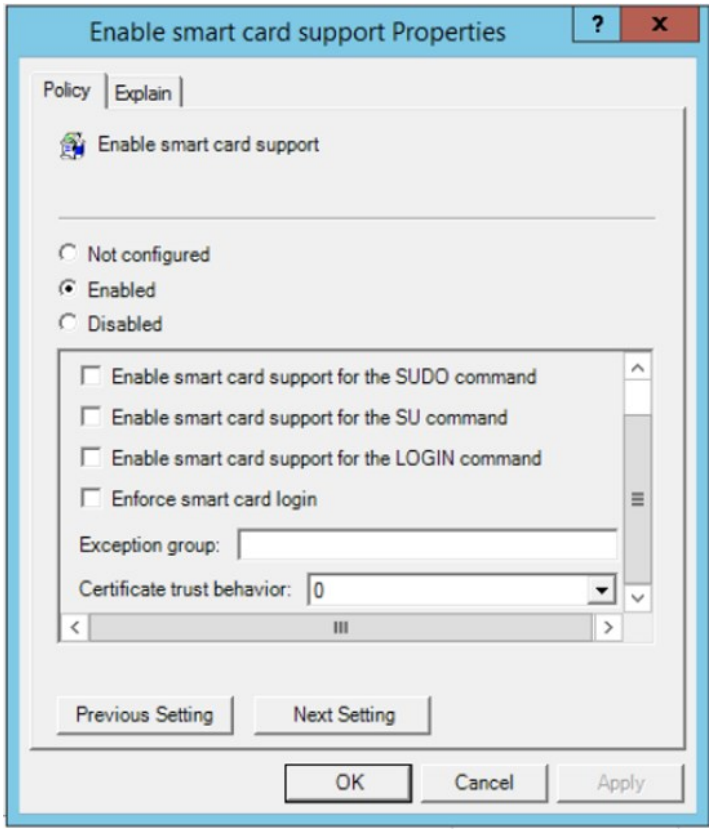
- Make sure that your smart card is able to log in to a Windows computer.

  If a user is able to log in to a Windows computer with a smart card, and you have a card reader and a fully-provisioned card for the Mac computer, the user should be able to log in to the Mac computer once you configure it for smart card support.

Enabling Smart Card Support

To enable smart card support for logging on

1. Make sure that you have configured the Delinea Agent to have full disk access.
2. Create or edit an existing Group Policy Object linked to a site, domain, or OU that includes Mac computers.
3. In the Group Policy Management Editor, expand **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy**, then double-click **Enable smart card support**.



4. Select **Enabled** to enable smart card support.
5. Select any of the following smart card options:
    - **Enable smart card support for the SUDO command**: When executing the SUDO command, the smart card user can authenticate by entering their smart card PIN.
    - **Enable smart card support for the SU command**: When executing the SU command, the smart card user can authenticate by entering their smart card PIN.
    - **Enable smart card support for the LOGIN command**: When executing the LOGIN command, the smart card user can authenticate by entering their smart card PIN.
    - **Enforce smart card login**: Users can only log in to the Mac computer by way of smart card login.
    - **Exception group**: Any users who belong to this group can always log in to the Mac computer with user name and password (no smart card required). In general, we recommend that you set an exception group, such as admins, when you select the option to enforce smart card login.
    - **Certificate trust behavior**: You can select one of these numbers to set smart card certificate behavior. The numbers mean the following:
        - 0: Smart card certificate trust isn't required.
        - 1: Smart card certificate and certificate chain must be trusted.
        - 2: Certificate and certificate chain must be trusted and not receive a revoked status.
        - 3: Certificate and certificate chain must be trusted and revocation status is returned valid.
6. Because smart card login is not password-based, **do not** enable the "Enable Keychain synchronization" group policy: Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy › Enable Keychain synchronization
7. If FileVault is enabled on your Mac, please enable the "Disable automatic login" group policy: Computer Configuration › Policies › Centrify Settings › Mac OS X Settings › Security & Privacy › FileVault2 › Disable automatic login.

   The policy takes effect dynamically at the next group policy refresh interval or after you run adgpupdate.

Verifying Smart Card Configuration

After enabling smart card support as described above in Configuring Smart Card Login, do the following to verify that a smart card is working:

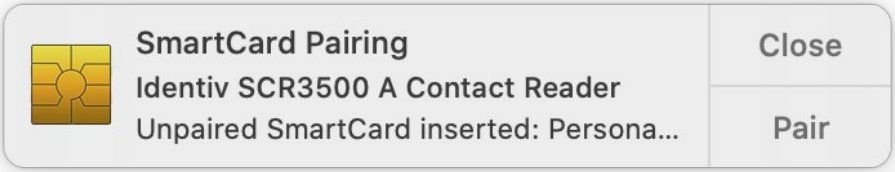
1. Insert the smart card into the reader.
2. Open the Terminal.app and run the following command:

   ```
   % sc_auth identities
   ```

   You should see that the smart card has paired to the Active Directory user. For example:

```
SmartCard: com.apple.pivtoken:00000000000000000000000000000000
Paired identities which are used for authentication:
9800A35AD2A41AEFB03CF431B76BA194E22F48EE    pivau1 - Certificate For PIV Authentication (PIV AU 1)
```

Note: You never need to pair your smart card manually. If you see the following SmartCard Pairing dialog, that means that the smart card support is not ready. Please re-check the smart card support GP and then excute the command adgpupdate.



Enabling the Sscreen Saver for Smart Card removal

Currently, we don't have a group policy to enable the screen saver when the smart card is removed. Please use the group policy entitled "Specify multiple login scripts" to deploy the following script:

```
#!/bin/bash
user_name=`ls -l /dev/console | cut -d " " -f 4`
defaults write /Users/$user_name/Library/Preferences/com.apple.screensaver tokenRemovalAction -int 1
chown $user_name:staff /Users/$user_name/Library/Preferences/com.apple.screensaver.plist
exit 0
```

The script sets the Mac to start the screen saver automatically when the smart card is removed.

Disabling Smart Card Support

To disable smart card support:

1. Edit the Group Policy Object linked to a site, domain, or OU that includes Mac computers, expand **Computer Configuration > Policies > Centrify Settings > Mac OS X Settings > Security & Privacy**, then double-click **Enable smart card support**.
2. Select **Disabled** and click **OK**.

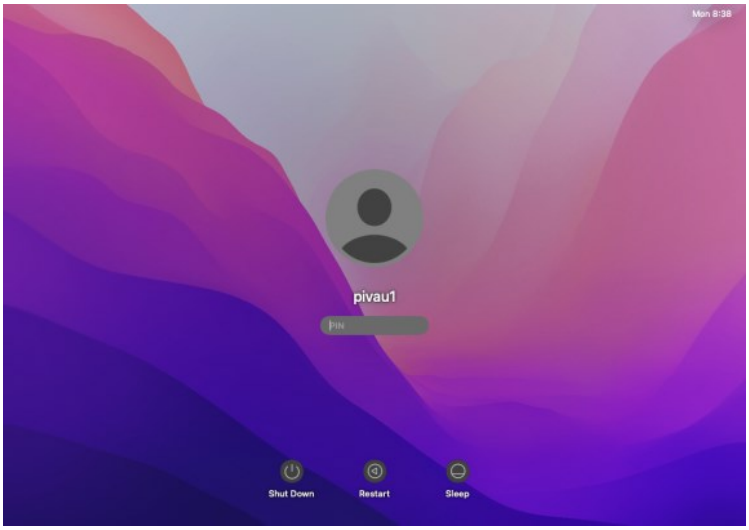
Using Smart Card Login

When a user inserts a smart card into the card reader attached to a Mac computer that is waiting for login, the login screen is replaced by a smart card enabled login screen.

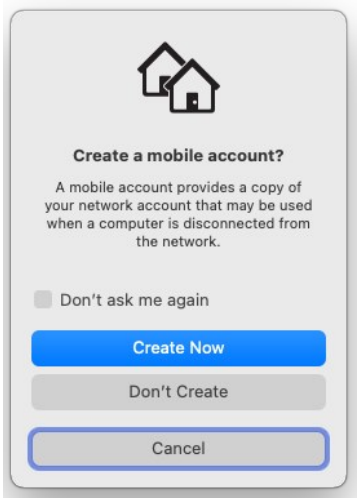
To log in with a smart card:

1. Insert the smart card into the smart card reader.

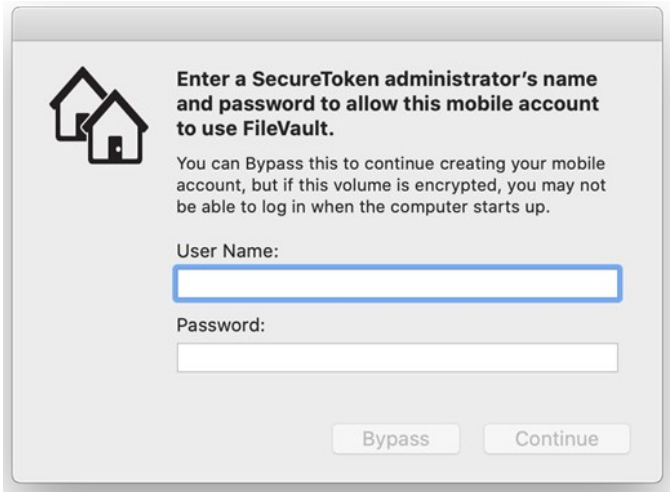
   A login screen displays, prompting you to enter your PIN.



2. If the "Configure mobile account creation" group policy is enabled, you are prompted to create a mobile account.



If the **Bypass the SecureToken dialog** is not enabled, after creating the mobile account, you are prompted to authenticate the SecureToken.



3. The system will then prompt you to set a password for Keychain.

   The password can be the same as or different than your Active Directory password. For security reasons, the password here should not be the same as your smart card PIN.

Troubleshooting Smart Card Login

If you have problems with smart card logon, Access Manager provides a command-line tool, sctool, which you can run to configure smart card logon, as well as to provide diagnostic information. See the sctool man page.

Additional smart card diagnostic procedures are provided in Diagnosing Smart Card Login Problems.

Other Functions of Smart Card Support on MacOS

MacOS 10.15 includes built-in support for the following capabilities:

- Authentication: LoginWindow, PKINIT, SSH, Screensaver, Safari, authorization dialogs, and in third-party apps supporting CTK
- Signing: Mail and third-party apps supporting CTK
- Encryption: Mail, Keychain Access, and third-party apps supporting CTK

For more information, please see https://support.apple.com/guide/deployment-reference-macos/intro-to-smart-card-integration-apd1fa5245b2/1/web/1.0.

Known Issues of Using Smart Cards with Macos

Due to a limitation of MacOS, a smart card user cannot get the User Group Policy automatically. The Computer Group Policy works normally. The workaround is to run the following commands after logging in with a smart card:

```
% sctool -k

% adgpupdate
```

After you run the above commands, log out and log back in. Most User Group Policy should work normally.

This section provides troubleshooting tips for administrators using the Delinea DirectControl Agent for Mac.

The following topics are covered:

Using Common Account Management Commands

Most UNIX-based platforms store account information in the local /etc/passwd file, and use commands such as getent command to query that information. On Mac computers, however, you would typically use the Directory Service application to manage local accounts and retrieve user information. For troubleshooting purposes, therefore, you should be familiar with the commands to use for retrieving information about Active Directory users and groups.

The following table describes several common Directory Service Command Line (dscl) commands that you may find useful.

dscl /Search –list /Users	List all of the users in the Directory Service and in Active Directory for the zone.
dscl /CentrifyDC –list /Users	List only the Active Directory users enabled for the zone.
dscl /CentrifyDC –read /Users/username	Display detailed information about the specified Active Directory *username*.
dscl /Search –list /Groups	List all of the groups in the Directory Service and in Active Directory for the zone.
dscl /CentrifyDC –list /Groups	List only the Active Directory groups enabled for the zone.
dscl /CentrifyDC –read /Groups/groupname	Display detailed information about the specified Active Directory *groupname*.

To get detailed information for all users or groups recognized on the Mac computer, you can use the following commands:

```
lookupd –q user –a name
```

```
lookupd –q group –a name
```

To get detailed information for a specific user or group, you can use the following commands:

```
lookupd –q user –a name username
```

```
lookupd –q group –a name groupname
```

To clear the Directory Service cache, you can use the following command:

```
lookupd -flushcache
```

To completely clear the cache of Active Directory login credentials, you should also run the `adflush command:

```
adflush
```

To retrieve Mac OS version and build information that `uname -a` does not provide, you can run the following command:

```
/usr/bin/sw_vers
```

Viewing the Agent Version on the Macs Joined to Active Directory

You can use the Active Directory module for Windows PowerShell to view the version of the Delinea DirectControl Agent for Mac on the Macs joined to your AD domain. This is useful to verify that all Macs joined to your AD have an appropriate version of the Delinea DirectControl Agent for Mac to avoid compatibility issues with OS updates.

Install the Active Directory Module for Windows PowerShell

The Active Directory module for Windows PowerShell is already installed on domain controllers. If you are using a member server, you will have to install it.

To install the Active Directory module for Windows PowerShell

Open an elevated PowerShell session on a Windows server in the domain and run the following command:

```
Add-WindowsFeature RSAT-AD-PowerShell
```

When the installation finishes it returns the following:

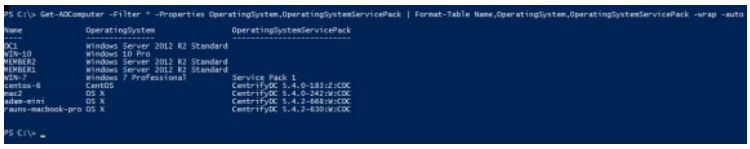
Once installed, on Windows Server 2012 and 2012 R2 the module automatically loads when you use one of its cmdlets; you do not need to import it.

Show PowerShell Output of Agent Versions for AD-joined computers

If you have a small environment, or just want to see a sample of the information that will be in the report, run the following from a Windows server with the AD PowerShell module installed:

```
Get-ADComputer -Filter * -Properties OperatingSystem,OperatingSystemServicePack | Format-Table Name,OperatingSystem,OperatingSystemServicePack -wrap -auto
```

For example:



The report includes all AD-joined computers in the domain. The example above shows a mix of Windows, Linux, and Mac computers. Where OperatingSystemServicePack is empty, it means there is no Service Pack installed (Windows computers), or there is no DirectControl agent installed (Mac or Linux/Unix).

In most cases there are too many computers for the PowerShell output to be easily readable. In these cases, refer to the next section, **Export the Report of Agent Versions to a CSV File**.

Export the Report of Agent Versions to a CSV file

You can export a report of Delinea DirectControl Agent for Mac versions on AD-joined computers to a CSV file for easier manipulation by running the following:

```
Get-ADComputer -Filter * -Properties OperatingSystem,OperatingSystemServicePack | Select-Object Name,OperatingSystem,OperatingSystemServicePack | Export-CSV CDCVersion.csv -NoTypeInformation -Encoding UTF8
```

In this example, PowerShell exports the data described above in **Show PowerShell Output of Agent Versions for AD-joined Computers** to a CSV file named CDCVersion.csv in the current directory. You can then open that CSV file using a spreadsheet application such as Excel to more easily analyze the data.

Enabling Logging for the Delinea DirectControl Agent for Mac

The Delinea DirectControl Agent for Mac installation includes some basic diagnostic tools and a logging mechanism to help you trace the source of problems if they occur. These diagnostic tools and log files allow you to periodically check your environment and view information about the agent operation, your Active Directory connections, and the configuration settings for individual computers.

In most cases, logging is not enabled by default for performance reasons. Once enabled, however, log files provide a detailed record of Delinea DirectControl Agent for Mac activity and can be used to analyze the behavior of Delinea Management Services and communication with Active Directory to locate points of failure.

For performance and security reasons, you should only enable agent logging when necessary, for example, when requested to do so by Delinea Corporation Technical Support, and for short periods of time to diagnose a problem. Keep in mind that sensitive information may be written to this file and you should evaluate the contents of the file before giving others access to it.

You can enable logging either by using the cdcdebug command or the Delinea for Mac Diagnostic Tool application.

To enable logging with the cdcdebug command:

1. Log in to the Mac as Local Admin and open the Terminal.
2. Run the following commands to clear and then enable the Delinea DirectControl Agent for Mac log file:

   ```
   % sudo /usr/local/share/centrifydc/bin/cdcdebug clear
   ```

   ```
   % sudo /usr/local/share/centrifydc/bin/cdcdebug on
   ```

3. Record the start time point:

   ```
   % date +%s
   ```

   For example: the output is 1610614011, please remember this output, it is the start time point.
4. Log out of the local admin user account.
5. Reproduce the issue: try to log in as the affected Active Directory user. Let it fail.
6. Log back in as Local Admin and open the Terminal again.
7. Record the end time point:

   ```
   % date +%s
   ```

   For example: the output is 1610614043, please remember this output, it is the end time point)
8. Enter the following commands to collect the Delinea DirectControl Agent for Mac log file:

   ```
   % sudo /usr/local/share/centrifydc/bin/cdcdebug -f pack [affected_AD_user_name] [start_time_point] [end_time_point]
   ```

   ```
   % adquery user -A [affected_AD_user_name] > /tmp/adquery.log
   ```

9. Send us the following files for analysis:

   ```
   /var/centrify/tmp/cdcdebug.tar.gz
   ```

   ```
   /tmp/adquery.log
   ```

10. Disable the Delinea DirectControl Agent for Mac log:

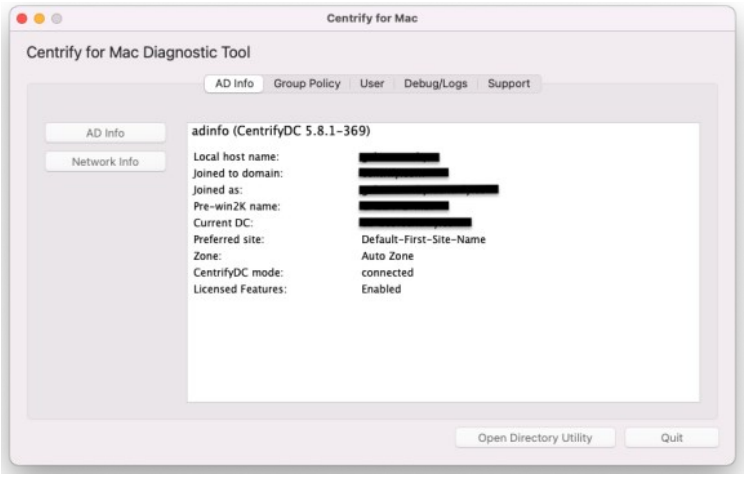
    ```
    % sudo /usr/local/share/centrifydc/bin/cdcdebug off
    ```

To enable logging with the Delinea for Mac Diagnostic Tool:

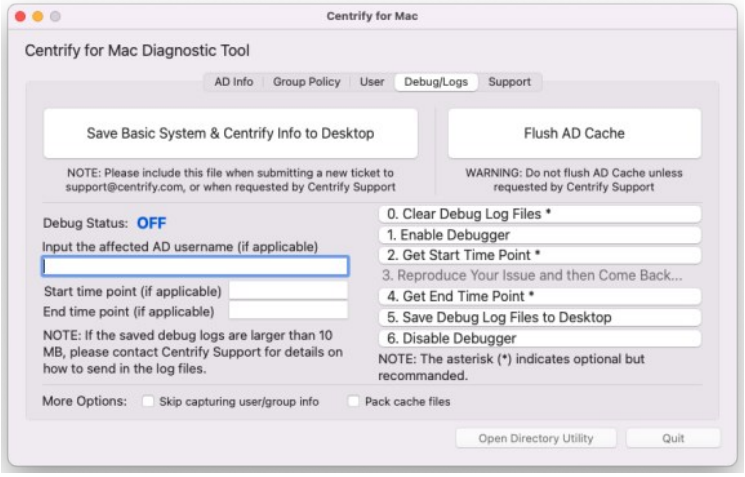
1. Log in to the Mac as Local Admin and open the application MacDiagnosticTool.app.

   The location of this app is "/Library/Application Support/Centrify/MacDiagnosticTool.app." You can run the following command to open it:

   ```
   % open /Library/Application Support/Centrify/MacDiagnosticTool.app
   ```

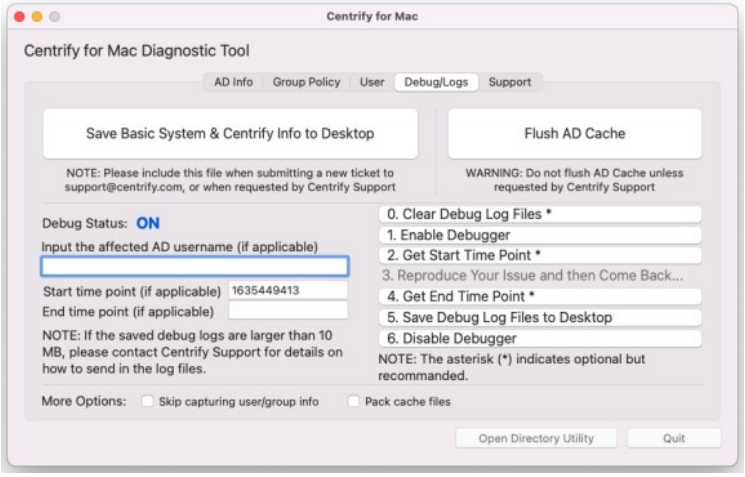


2. Click the **Debug/Logs** tab.

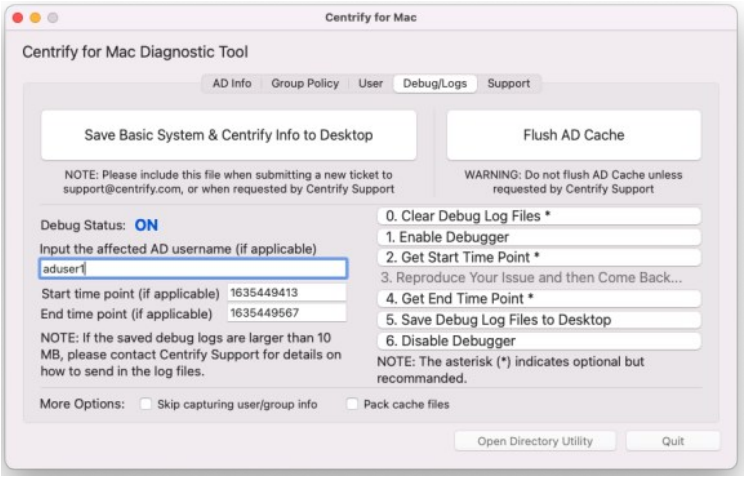


3. Click **0. Clear Debug Log Files**.
4. Click **1. Enable Debugger**.
5. Click **2. Get Start Time Point**.

   Note: You do not need to remember the start time point; it will be saved automatically.

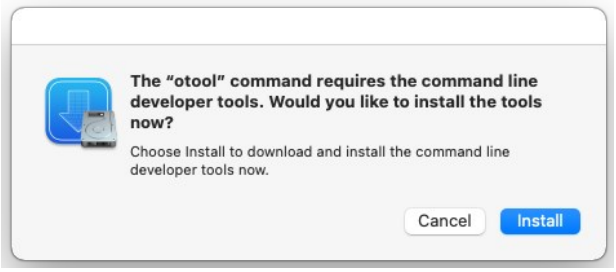


6. Click **Quit** to close the application.
7. Log out of the Local Admin account.
8. Reproduce the issue: try to log in as the affected Active Directory user. Let it fail.
9. Log back in as Local Admin and open the application MacDiagnosticTool.app again.
10. Click **4. Get End Time Point** and enter input the affected Active Directory user name.



11. Click **5. Save Debug Log Files to Desktop**, the tool will start to collect agent log files.

    Note: You might see a message display about installing the "otool" command; you can select **Cancel** or **Install**; either choice works.



The log file "CENTRIFY_FULL_LOG_PACK.zip" will be on the Desktop. Send the file to Delinea Technical Support for analysis.

12. Click **6. Disable Debugger**, then click **Quit** to close the application.

Enabling Logging for the Mac Directory Service

In addition to enabling logging for the agent, you may find it necessary to enable logging for the Open Directory Service.

To create a log file for the Open Directory Service:

1. Log in as or switch to the root or admin user.
2. Run the following command:

```
odutil set log debug
```

After running this command, you can find the resulting log files at: /var/log/opendirectoryd.log*. You can then provide both the agent log file and the Directory Service log file to Delinea Support if you need assistance troubleshooting issues.

Using the Agent on a Dual-boot System

If you are using a dual-boot system, and the computer name is the same for each version of the operating system, the Delinea DirectControl Agent for Mac(adclient) will not launch when you reboot and switch operating systems. The problem is that each operating system sets its own password for adclient and the password does not work for the other operating system.

The best way to avoid this problem is to provide a different computer name for each operating system. Because the computer names are different, the password for one operating system is not changed by the other operating system.

If you want to use the same computer name for both operating systems, you can work around the problem, as follows:

1. Leave the domain (adleave) before rebooting and switching operating systems.

   Note: You may leave and join the domain after rebooting and switching the operating system. However, you will experience some delay while adclient attempts to launch and fails.

2. Reboot with the other operating system.
3. Rejoin the domain (adjoin).

Using adgpupdate Appropriately

If adgpupdate is run multiple times in succession, it is possible that not all group policies will be applied correctly. To avoid this problem, do not run adgpupdate more than once per minute.

Understanding Delays when Logging on the First Time with a New User Account

Depending on the configuration of your startup services, you may find that new users are unable to log on to a computer immediately (within the first 15 to 30 seconds) after a computer is rebooted.

By default, the Mac login window only requires the DiskS and SecurityService startup services to start successfully to prompt for the user to log in. Authenticating users to Active Directory, however, requires the additional DirectoryServices startup service to be available. Starting the DirectoryServices startup service causes a 10 to 15 second delay before the LoginWindow can successfully authenticate new Active Directory users.

Configuring Single-sign on to Work with Non-Mac Computers

On a Mac computer, the ssh client does not forward (delegate) credentials to the server by default. Therefore, when attempting to use ssh from a Mac computer with DirectControl agent installed to a non-Mac computer with DirectControl agent installed, single sign-on (SSO) does not work. To fix this problem, set the configuration parameter, GSSAPIDelegateCredentials, to yes in the /etc/ssh_config file on the Mac computer.

Restricting Login Using FTP

In Active Directory, you can set properties to prevent a user from logging in to other Macintosh computers. However, this restriction will not prevent a user from logging in via FTP to Macintosh computers with the DirectControl agent installed. It does restrict logging in with telnet, ssh, rlogin, and rsh.

Logging on Using localhost

For many UNIX platforms, you can log on using localhost to refer to the local computer; for example:

```
root@localhost
```

This syntax does not work when logging on to a Macintosh computer, whether using the Macintosh UI, or remotely through ssh or FTP.

Changing the Password for Active Directory Users

In the Mac OS X, the passwd command authenticates the user only after you type the user password. Because of this, the passwd command does not recognize the user as an Active Directory user until after the password is entered and the password prompts defined for Active Directory users, which are typically set through group policy or by modifying the Delinea configuration file, are not displayed. You can still use the passwd or chpass command to change the Active Directory password for a user, but you will not see any visual indication that you are modifying an Active Directory account rather than a local user account.

Disabling the Apple Built-in Active Directory Plug-in

Apple provides a built-in Apple Directory plug-in that may interfere with the Delinea DirectControl Agent for Macinstallation and operation. Therefore, before installing the agent, disable Apple's built-in Active Directory plug-in. In addition, remove Active Directory from the Authentication and Contacts search paths. If this plug-in is enabled and the Delinea DirectControl Agent for Mac has been installed, disable the plug-in, then reboot the Macintosh computer for reliable operation.

To disable the Apple Directory plug-in and remove Apple Directory from the Authentication and Contacts search paths:

1. On a Mac computer, open the Directory Utility.

   You can find the Directory Utility in one of these folders depending on the operating system that you are running:

   - /System/Library/CoreServices
   - /Applications/Utilities

2. Click the lock icon and enter credentials to allow you to make changes.
3. Click the **Search Policy** icon.
4. Click the **Authentication** tab, then select Custom path in the **Search** box.

   If Active Directory was previously enabled, Active Directory appears in the Directory Domains box; for example:

   ```
   /Active Directory/All Domains
   ```

5. Select **/Active Directory/All Domains** and click **Remove** — or select the minus - sign). Then click **Apply**.
6. Click the **Contacts** tab, then select Custom path in the **Search** box. If Active Directory was previously enabled, Active Directory shows (in red font) in the Directory Domains box; for example:

   ```
   /Active Directory/All Domains
   ```

7. Select **/Active Directory/All Domains** and click **Remove**. Then click **Apply**.
8. Close the window.
9. If you have already installed the Delinea DirectControl Agent for Mac, reboot the computer.

Showing the Correct Status of the Delinea Plug-in

The Delinea plug-in is automatically added to the list of Apple Directory Utility plug-ins that are used for lookup and authentication. However, if the Apple Directory Utility tool is running when you install the Delinea DirectControl Agent for Mac, or when you join or leave a domain before updating to a new version of the agent, it will incorrectly display the status of the plug-in. For example, it will show the status as disabled, when in fact, the plug-in is enabled.

To avoid this problem, before launching the installer, be certain that the Apple Directory Utility tool is closed.

If the Directory Utility was open during installation, simply close and re-open Directory Utility, then make certain that the Delinea plug-in is enabled.

You may also restart the Delinea plug-in from the command line, as follows:

1. Close the Directory Utility.
2. Open a terminal.
3. Enter the following command:

   ```
   /usr/local/share/centrifydc/bin/dsconfig restart
   ```

4. Open the Directory Utility. The status of Delinea should be enabled.

Resolving VPN Access Issues with Mac OS X 10.7 and Later

Starting with Mac OS X 10.7, /etc/resolv.conf is no longer used for domain controller name resolution. Therefore, some VPN programs no longer update DNS server information in /etc/resolv.conf when signing on. On computers running Mac OS X 10.7 and later, this can result in the computer not being able to connect to a domain controller through a VPN.

To resolve this issue, explicitly specify in centrifydc.conf the location of DNS servers that are used to resolve domain controller names:

1. Open /etc/centrifydc/centrifydc.conf for editing.
2. Specify the IP addresses of DNS servers in the dns.servers parameter (if the parameter does not exist yet, create it now):
   ```
   dns.servers: x.x.x.x y.y.y.y
   ```

   where x.x.x.x y.y.y.y are the IP addresses of the DNS servers to use. This example shows two IP addresses; note that each IP address is separated by a space.
3. Save your changes to centrifydc.conf.
4. Restart the agent for the changes to take effect:

   ```
   sudo /usr/local/share/centrifydc/bin/centrifydcrestart
   ```

Diagnosing Smart Card Login Problems

Two general methods for diagnosing smart card log in problems are provided:

- By using the sctool utility as described in the sctool man page.
- By performing the diagnostic procedures described in this section.

The following procedures are intended to diagnose multiple causes of smart card log in failure. It is recommended that you retest smart card login at regular intervals (such as after each step) as you perform this procedure.

1. Ensure that macOS built-in PIV token is not disabled.

   ```
   % defaults read /Library/Preferences/com.apple.security.smartcard DisabledTokens
   ```

   It should not exist.
2. Ensure that smart card support is enabled.

   ```
   % sctool -s
   ```

   It should show that smart card support is enabled.
3. Ensure that the smart card can be recognized by MacOS.

   ```
   % sc_auth identities
   ```

   It should show your card and the card has been paired to the Active Directory user.
4. Collect support information.

```
% sctool -S
```

Send the file /tmp/sctool.support to Delinea Support.

Opening a Support Case Online

If you need assistance with troubleshooting an issue, you may need to open a case with Delinea Support. Before opening a new case, Delinea recommends searching the Delinea Support Portal to see if your problem is a known issue or something for which there is a recommended solution.

To search the Delinea Support Portal

1. Open https://www.delinea.com/support/ in a Web browser.
2. Click in the search field and type one or more key words to describe the issue, then click the search icon to view potential answers to your question.

If your issue is not covered in one of the search results, you should open a case with Delinea Support.

To open a new support case

1. Log in to the Delinea support portal.
2. Click **Manage Cases**, then click **Open a New Support Case**.

   The NEW CASE DETAIL page appears.
3. Enter your case details, then click **Next**.

   Provide as much information as possible about your case, including the operating environment where you encountered the issue, and the version of the Delinea product you are working with.

   A new page appears showing Suggest Knowledge Articles and Technical Resources. You can click **Show More** to see additional resources that might solve your problem.
4. Click **No Thanks, Submit a Case** to open a new case.

Collecting Information for Support Cases

To help ensure your issue gets resolved quickly and efficiently, gather as much information about your working environment as possible. See the information below in these two sections:

- Collecting general information about your environment
- Collecting information specific to login events

Collecting Information Specific to Smart Card Login Failure

Collect the following information prior to opening a support case related to smart card log in failure:

- The smart card type (for example, PIV, CAC, CACNG, and so on), manufacturer, and model.
- A screen image of the smart card and its certificates in Keychain Access.
- The following log files:

  /tmp/sctool_D.log

  /tmp/adquery.log

  /tmp/tokendfolder.log

  /var/Centrify/tmp/adinfo_support.tar.gz

To generate these logs, run the following commands while logged in as the local administrator:

```
sctool -D > /tmp/sctool_D.log

adquery user -A username_of_smartcard_user > /tmp/adquery.log

sudo ls -l /System/Library/Security/tokend/ > /tmp/tokendfolder.log

sudo adinfo -t
```

Collecting General Information about Your Environment

Take the following steps to gather information about your working environment before opening a support case.

1. Verify that the DirectControl agent is running on the computer where you have encountered a problem. For example, run the following command:

   ```
   ps aux | grep adclient
   ```

   If the adclient process is not running, check whether the watchdog process, cdcwatch, is running:

   ```
   ps aux | grep cdcwatch
   ```

   The cdcwatch process is used to restart adclient if it stops unexpectedly.

   > **Note**: The commands in the following three steps must be run as root or with the sudo command.

2. Enable logging for the DirectControl agent; for example:

   ```
   sudo /usr/local/share/centrifydc/bin/cdcdebug on
   ```

   > **Note**: Login events are captured in /var/log/centrifydc-login.log by default. Turning on cdcdebug captures login events in /var/log/centrifydc.log.

3. Create a log file for the Mac Directory Service. For example:

   - To enable logging for opendirectoryd:

   ```
   odutil set log debug
   ```

   - To disable logging for opendirectoryd when sufficient log information is collected:

   ```
   odutil set log default
   ```

4. Duplicate the steps that led to the problem you want to report. For example, if an Active Directory user can't log in to a managed system, attempt to log the user in and confirm that the attempt fails. Be sure to make note of key information such as the user name or group name being used, so that Delinea Support can identify problem accounts more quickly.

5. Verify that log file /var/log/centrifydc.log or /var/adm/syslog/centrifydc.log exists and contains data.

6. Run the cdcdebug command to generate logs that describe the domain and current environment; for example:

   ```
   sudo /usr/local/share/centrifydc/bin/cdcdebug -f pack username
   ```

   The following log files are created in /var/centrify/tmp when you execute the cdcdebug command:

   - adinfo_support.tar.gz
   - adinfo_support.txt
   - cdcdebug.tar.gz
   - dump_cache_error.log
   - stacktrace.txt

7. If there is a core dump during or related to the problem, save the core file and inform Delinea Support about it. Delinea Support may ask for the file to be uploaded for review.

   If the core dump is caused by a Delinea process or command, such as adclient or adinfo, open the /etc/centrifydc/centrifydc.conf file and change the adclient.dumpcore parameter from never to always and restart the agent:

   ```
   sudo /usr/local/share/centrifydc/bin/centrifydcrestart
   ```

   > **Note**: For more information about starting and stopping the agent, see the *Administrator's Guide for Linux and UNIX*.

8. If there is a cache-related issue, Delinea Support may want the contents of the /var/centrifydc directory. You should be able to create an archive of the directory, if needed.
9. If there is a DNS, LDAP, or other network issue, Delinea Support may require a network trace. You can use Ethereal to create the network trace from Windows or UNIX. You can also use Netmon on Windows computers.
10. Create an archive (for example, a .tar or .zip file) that contains all of the log files and diagnostic reports you have generated, and add the archive to your case or send it directly to Delinea Support.
11. Consult with Delinea Support to determine whether to turn off debug logging. If no more information is needed, run the following commands, which must be run as root or with sudo:

    ```
    odutil set log default

    sudo /usr/local/share/centrifydc/bin/cdcdebug off
    ```

Collecting Information Specific to Login Events

Login events are captured in /var/log/centrifydc-login.log by default. If you enable logging for the DirectControl agent by turning on cdcdebug, login events are then captured in /var/log/centrifydc.log.

The /var/log/centrifydc-login.log grows to a maximum size of 50M before it is compressed. When all compressed centrifydc-login.log files combined with the current log file exceed 250M, the oldest compressed log is replaced.

This section shows other methods of installing the agent besides the standard method using the package installer (DMG file); see Installing the Delinea DirectControl Agent for Mac. It also shows how to remove the agent and how to join and leave a domain.

The following topics are covered:

Installing Using the install.sh Script

Installing Silently on a Remote Computer

Uninstall from the Delinea System Preferences Pane

Run the uninstall.sh Script

Leaving an Active Directory Domain

Installing Using the install.sh Script

This section explains how to install using the install.sh script. This method is recommended for experienced UNIX administrators who are familiar with UNIX command-line installations. Otherwise, you should install by using the graphical user interface, which is described in Installing the Delinea DirectControl Agent for Mac.

To install using the install.sh command-line program:

> **Note**: Before launching the installer, be certain that Apple Directory Utility is closed. If it is open while running the installer, it causes the Delinea Directory Access plug-in to show the incorrect status, that is, it shows that the plug-in is disabled when in fact it is enabled.

1. Log on with a valid user account.

   > **Note**: You are not required to log on as the root user on, but you must know the password for the Administrator account to complete the installation.

2. Mount the CD-ROM device using the appropriate command for the local computer's operating environment, if it is not automatically mounted.
3. Change to the appropriate directory on the CD or on the network where the DirectControl agent package is located. For example, change to the Agent_Mac directory.
4. Run the install.sh script to start the installation of the Delinea software on the local computer's operating environment. For example:

   ```
   sudo ./install.sh
   ```

   Before beginning the installation, the install.sh script runs the ADCheck utility, which performs a set of operating system, network, and Active Directory checks to verify that the Mac computer meets the system requirements necessary to install the Delinea DirectControl Agent for Mac and join an Active Directory domain.

5. Review the results of the checks performed. If the target computer, DNS environment, and Active Directory configuration pass all checks with no warnings or errors, you should be able to perform a successful installation and join. If you receive errors or warnings, correct them before proceeding with the installation.
6. Follow the prompts displayed to select the services you want to install and the tasks you want to perform. For example, you can choose whether you want to join a domain or restart the local computer automatically at the conclusion of the installation.

When installation is complete, see **Understanding the Directory Structure** below for a description of the directories and files installed for Centrify.

Installing Silently on a Remote Computer

You can install the agent silently on a remote Mac computer in either of these ways:

- By using sudo commands from the command line. If you use this method, no user interaction on the target Mac computer is required. See the section below, **Installing Remotely on a Mac Computer Using sudo Commands**.
- By using Apple Remote Desktop. This method requires that you have Apple Remote Desktop 3 for remote software distribution. See the section below, **Installing Remotely on a Mac Computer Using Apple Remote Desktop**.

If you use this method to install version 5.1.0 of the agent, the Delinea Join Assistant launches on the target Mac computer after the installation completes, and a user must interact with the Delinea Join Assistant to complete the join process. This limitation exists only in version 5.1.0 of the agent. Earlier versions of the agent (that is, 5.0.x and lower) and later versions (5.1.1 and above) do not have this limitation, and can be installed using Apple Remote Desktop without any user interaction on the target Mac computer.

Installing Remotely on a Mac Computer Using Sudo Commands

Perform the following steps to use sudo commands to install the agent remotely on a target Mac computer without requiring any user interaction on the target Mac computer.

To install the agent remotely using sudo commands:

1. Ensure that you have administrator account credentials on the target Mac computer, and that SSH is installed on the target Mac computer.
2. On the computer where the Delinea packages were downloaded (that is, the source computer), use an appropriate file transfer method to push the CentrifyDC-x.x.x.pkg file to the target Mac computer.

   For example, perform these steps to transfer files from a PC source computer to the target Mac computer:

   1. On the source computer, ensure that file sharing is enabled, and that the folder containing the Delinea packages is a shared folder.
   2. On the target Mac computer:
      1. Open a new window in the Finder.
      2. In the sidebar under **Shared**, click **All**.
      3. Select the source computer.
      4. Click **Connect As**, type the user name and password for the source computer, and click **Connect**.
   3. The folder that you shared on the source computer appears in the Finder on the target Mac computer. Locate the CentrifyDC-x.x.x.pkg file on the source computer and drag it to the location of your choice on the target Mac computer.
3. On the source computer, use a program such as Putty to connect remotely to the target Mac computer through SSH. Log in to the target Mac computer using an account that has local administration privileges, such as the Local Admin account.
4. On the target Mac computer, navigate to the directory where the .pkg file was transferred and execute the following command:

   ```
   sudo /usr/sbin/installer –pkg CentrifyDC-x.x.x.pkg –target /
   ```

   When you execute this command, the agent is installed silently on the target Mac computer.

   - If an agent was already installed on the target Mac computer and this was an update of the existing agent, the target Mac computer was already joined to the domain, and you do not need to perform any additional steps.
   - If this was the first installation of the agent on the target Mac computer, you must enable licensed features and join the target Mac computer to a domain as described in Step 5 and Step 6.
5. Execute the following command on the target Mac computer to enable licensed features:

   ```
   sudo adlicense -l
   ```
6. When you join the target Mac computer to a domain, you can choose to join the auto zone or a specified hierarchical zone.

   - Execute the following command on the target Mac computer to join the target Mac computer to a domain and the Auto Zone:

     ```
     sudo /usr/local/sbin/adjoin --user Domain_Admin --container "domain.com/Path/To/OU"  --name computer_name --workstation domain_name.com
     ```

   - Alternatively, execute the following command on the target Mac computer to join the target Mac computer to a domain and a specified hierarchical zone:

     ```
     sudo /usr/local/sbin/adjoin --user Domain_Admin --container "domain.com/Path/To/OU"  --name computer_name --zone zone_namedomain_name.com
     ```

Installing Remotely on a Mac Computer Using Apple Remote Desktop

Perform the following steps to install the agent remotely on a target Mac computer without requiring any user interaction on the target Mac computer.

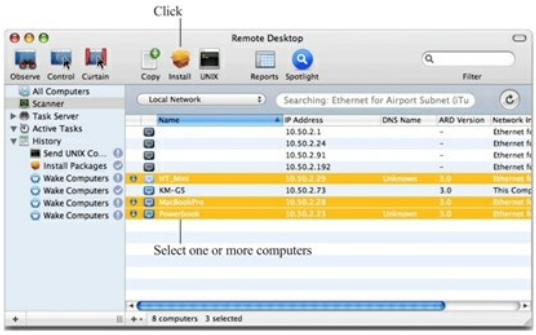
Note: If you use this method to install version 5.1.0 of the agent, the Delinea Join Assistant launches on the target Mac computer after the installation completes, and a user must interact with the Delinea Join Assistant to complete the join process. For all other versions of the agent, no user interaction on the target Mac computer is required.

To remotely install the DirectControl agent and join a computer to the domain using Apple Remote Desktop 3:

1. Verify that you have an Apple Remote Desktop 3 Admin station and one or more Apple Remote Desktop 3 Clients.
2. Verify that all of the Apple Remote Desktop 3 Client computers where you want to install the DirectControl agent are set to **Allow Remote Desktop** using the Service pane in the Sharing system preference. For example:

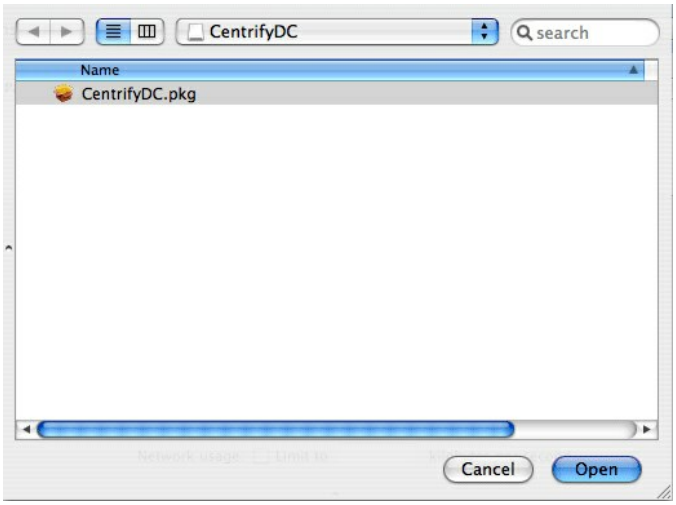
3. Copy the DirectControl agent package, for example centrifydc-release-macversion-i386.dmg, to the Apple Remote Desktop 3 Admin computer and verify that you can access the disk image.
4. Open Remote Desktop on the Admin Computer, then click **Scanner** and verify that the Mac computers on which you plan to install Delinea are listed and that ARD Version column displays 3.0 (or later). For example:

5. Select one or more computers from the list, then click **Install**. For example:

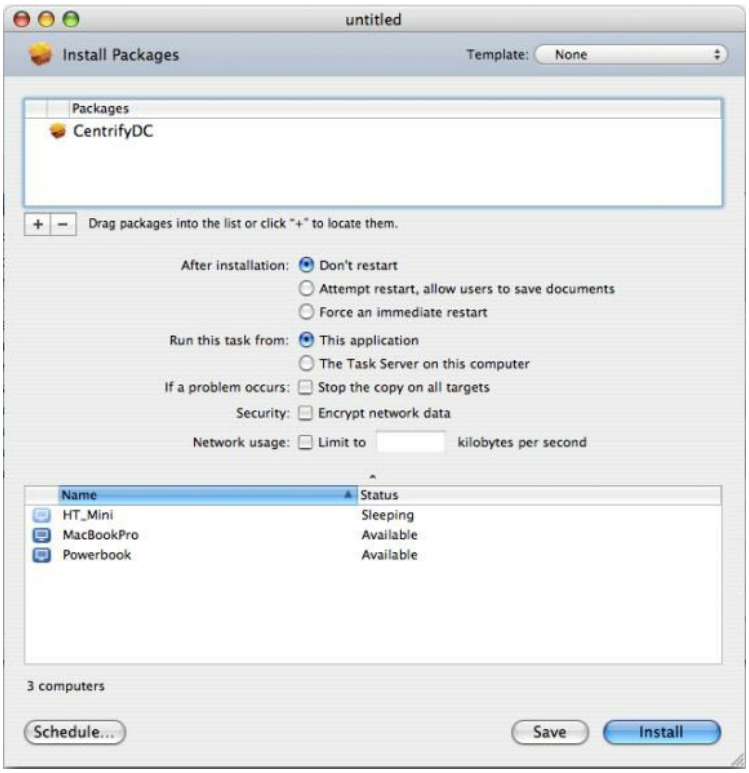


6. In the Install Packages window, click **+** to locate the CentrifyDC.pkg in the DirectControl agent disk image. For example:

7. In the DirectControl agent disk image, select the CentrifyDC.pkg file and click **Open** to add it to the Install Packages list. For example:



8. In the Install Packages window, click **Install** to install the listed packages, for example:



In most cases, you can use the default settings to install the Delinea DirectControl Agent for Mac. If you want to schedule the installation for another time rather than completing the installation now, click **Schedule**. For more information about the Apple Remote Desktop installation parameters, see Chapter 8 "Administering Client Computers," in the Apple Remote Desktop Manual.

If you click **Install** the Remote Desktop displays a progress bar and task status for each of the computers selected for the installation.

Understanding the Directory Structure

When you complete the installation, the local computer will be updated with the following directories and files:

/etc/centrifydc	The Delinea DirectControl Agent for Mac configuration file and the Kerberos configuration file.
/usr/local/share/centrifydc	Kerberos-related files and service library files used by the Delinea DirectControl Agent for Mac to enable group policy and authentication and authorization services.
/usr/local/sbin /usr/bin	Command line programs to perform Active Directory tasks, such as join the domain and change a user password.
/var/centrifydc	No files until you join the domain. After you join the domain, several files are created in this directory to record information about the Active Directory domain the computer is joined to, the Active Directory site the computer is part of, and other details.
/System/Library/Frameworks/ DirectoryService.framework/ Resources/Plugins	The Delinea Directory Service Plugin, CentrifyDC.dsplug, that enables you to join or leave the domain using the graphical user interface.

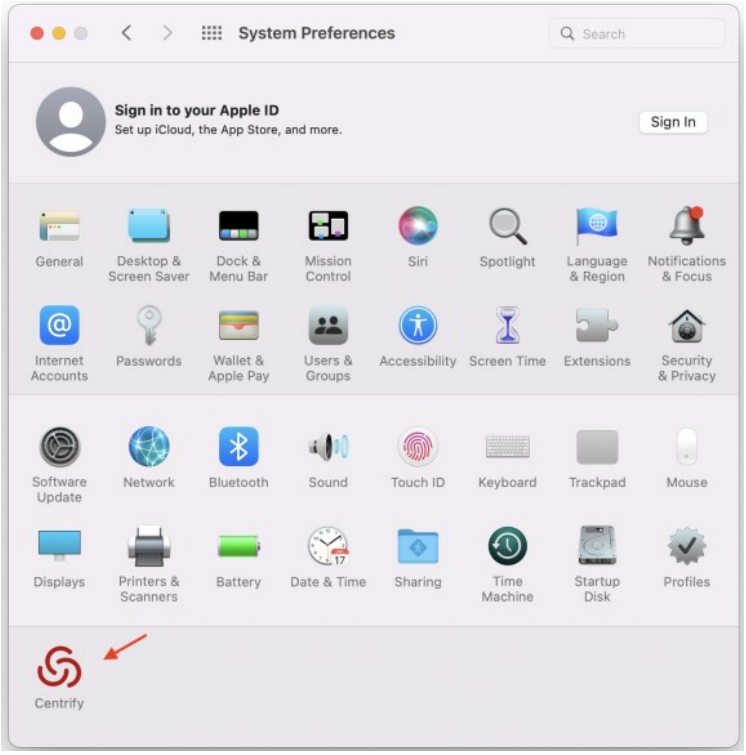
Uninstall from the Delinea System Preferences Pane

The Delinea System Preferences pane is created when you install the Delinea DirectControl Agent for Mac. You can use this pane to uninstall the Delinea

DirectControl Agent for Mac. Uninstalling the agent from the Delinea System Preferences pane also leaves the AD domain.

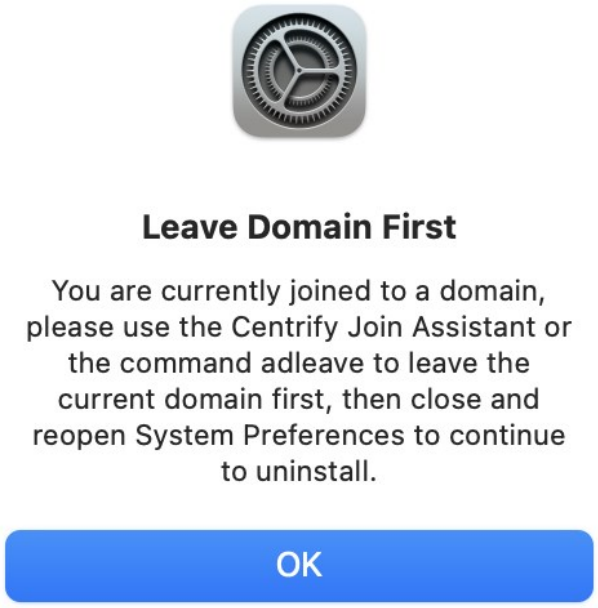
To uninstall the Delinea DirectControl Agent for Mac from the Delinea System Preferences pane

1. Open **System Preferences**, then click **Centrify**.



2. Click **Uninstall**, then click **OK** at the confirmation prompt.

   If you are currently joined to a domain, it will prompt the Leave Domain First dialog. For more information, see **Leaving an Active Directory Domain** below.



3. Enter administrator credentials and click **OK**.

   The uninstall process starts.

4. Click **OK** to quit when you see the window indicating that the Delinea DirectControl Agent for Mac was uninstalled.

Run the uninstall.sh Script

The uninstall.sh script is installed by default in the /usr/local/share/centrifydc/bin directory on each Centrify-managed system.

To remove the Delinea DirectControl Agent for Mac by running the uninstall.sh script

1. Open a Terminal window on the computer where the DirectControl agent is installed. For example, select **Applications > Utilities > Terminal**.
2. Switch to the root user or a user with superuser permissions. For example:

   ```
   su -
   ```

   Password: root_password

3. Run the uninstall.sh script. For example:

   ```
   /bin/sh /usr/local/share/centrifydc/bin/uninstall.sh
   ```

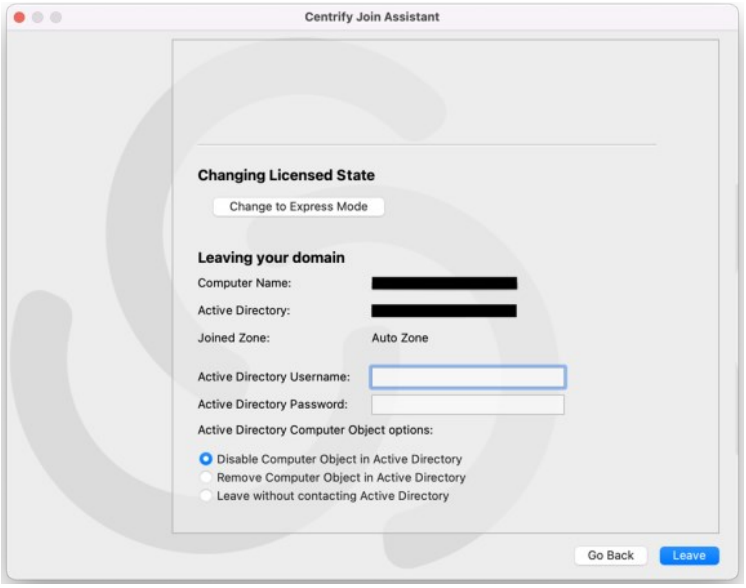
   The uninstall.sh script will detect whether the Delinea DirectControl Agent for Mac is currently installed on the local computer and whether the computer is currently joined to a domain. If the computer is not currently joined to a domain, the script will begin removing Delinea files from the local computer.

Leaving an Active Directory Domain

To start the Delinea program for joining or leaving a domain:

1. Click **Applications > Utilities > Delinea**, then double-click **Delinea Join Assistant** to open it.

   Click **Continue** on the Welcome page and the join assistant displays information about the domain to which the computer is connected:



2. Select whether to disable the computer object in Active Directory, remove the computer object from Active Directory, or leave without contacting Active Directory.

    - Disable: Disables the computer object in Active Directory.
    - Remove: Removes the computer object from Active Directory.
    - Leave without contacting Active Directory: This option forces the local computer's settings to their pre-join conditions without contacting Active Directory. The Computer Object will not be removed or disabled in Active Directory.

    Use this option if the Active Directory computer account has been modified or deleted so that the host computer can no longer work with it.

3. Click **Leave** to leave the domain.

The following topics are covered:

Server Suite is an IT management solution that provides three main services:

- Access control, provided through the Authentication Service.
- Privilege management, provided through the Privilege Elevation Service.
- Auditing, provided through Audit & Monitoring Service.

These services can be used together or independently, depending on the requirements of your organization.

Managing Windows Computers Using Delinea software

Server Suite is a security platform that includes multiple components for managing Windows computers. The components fall into two broad categories of features:

- Access-related components for managing access, including administrative privileges.
- Audit-related components for managing and analyzing audited activity.

Access-Related Features

Access-related features are provided by the Authentication Service and the Privilege Elevation Service. Together, these services enable you to manage access and administrative privileges for the computers in your organization. The primary tool for managing access-related features is Access Manager.

Access Manager provides a central console for defining and managing role-based access control rules and applying them to specific users, groups, or computers. For example, you can use Access Manager to delegate specific administrative tasks to a particular user or group. As an administrator, you can also use Access Manager to configure roles with start and expiration dates or limit the availability of a role to specific days of the week or hours of the day.

> **Note**: Server Suite treats gMSA accounts (group Managed Service Accounts) as Active Directory users.

Audit-Related Features

Audit-related features are provided by the Audit & Monitoring Service. This service enables you to collect and store audit trails that capture detailed information about user activity. The primary tool for managing audit-related features is Audit Manager.

Audit Manager provides a central console for configuring and managing audited computers, audit store databases, and the permissions granted to specific auditors. There is also a separate Audit Analyzer console for searching and replaying captured activity.

Choosing Access and Auditing Features

In addition to the management tools for access-related or auditing-related features, each computer you want to manage must have a Agent installed. After you install the agent, you choose whether to enable access features, auditing features, or both feature sets.

If you enable access features, the agent enforces the role-based privileges that enable users to run applications locally with administrative privileges without using the Administrator password and with their activity traceable to their own account credentials. You can also use role-based privileges to secure access to network services on remote computers.

If you enable auditing features, the agent captures detailed information about what users do when they access applications or network resources with administrative privileges.

You can use access features and components without auditing if you aren't interested in collecting and storing information about session activities. You can also deploy auditing features and components without access control and privilege elevation features if you are only interested in auditing activity on Windows computers. However, the real value of using Delinea software to manage Windows computers comes from using all of the services as an integrated solution for managing elevated privileges and ensuring accountability and regulatory compliance across all platforms in your organization.

Access Control for Windows Computers

By using Access Manager and deploying the Agent for Windows, you can develop fine-grained control over who has access to the Windows computers in your organization. You can also limit the use of administrative accounts and passwords. For example, you can restrict access to computers that host administrative applications or data center services and ensure that users accessing those computers can log on locally or connect remotely only when appropriate.

In a Windows environment without Delinea software, the primary way you secure access to Windows computers is by granting a limited number of users or groups local or domain administrator privileges. The main drawback of this approach is that the rights associated with group membership don't change. A user who has domain administrator rights has those rights on any computer in the domain at all times. In other cases, users who aren't administrators or members of an administrative group need administrative privileges to perform specific tasks that would require them to have an administrator and service account password. Shared passwords reduce accountability and are often flagged by auditors as a security issue.

Through the use of zones and roles, Delinea software provides granular control over **who** can do **what**, and over **where** and **when** those users should be granted elevated privileges.

One way trust environments

Windows agent supports one-way trust in the following scenarios:

- When the zone belongs to the resource forest.
- When the logon account belongs to the account forest.
- When the RunAs account or group belongs to the resource forest (RunAs group can be a built-in group).
- When the role assignment is at the zone, computer, or computer role level.

How Zones Organize Access Rights and Roles

One of the most important aspects of managing computers with Delinea software is the ability to organize computers, users, groups and other information about your organization into **zones**. A zone is a logical object created using Access Manager that is stored in Active Directory. You use zones to organize computers, rights, roles, security policies, and other information into logical groups. These logical groups can be based on any organizing principle you find useful. For example, you can use zones to describe natural administrative boundaries within your organization, such as different lines of business, functional departments, or geographic locations.

Zones provide the first level of refinement for access control, privilege management, and the delegation of administrative authority. For example, you can use zones to create logical groups of Windows computers to achieve these goals:

- Control who can log on to specific computers.
- Grant elevated rights or restrict what users can do on specific computers.
- Manage role definitions, including availability and auditing rules, and role assignments on specific computers.
- Delegate administrative tasks to implement "separation of duties" management policies.

You can also create zones in a hierarchical structure of parent and child zones to enable the inheritance of rights, roles, and role assignments from one zone to another or to restrict local or remote access to specific computers for specific users or groups.

Because zones enable you to grant specific rights to users in specific roles on specific computers, you can use zones as the first level of refinement for controlling who has access to which computers, where administrative privileges are granted, and time restrictions on when administrative privileges can be used.

You can also use zones to establish an appropriate separation of duties by delegating specific administrative tasks to specific users or groups on a zone-by-zone basis. With zones, administrators can be given the authority to manage a given set of computers and users without granting them permission to perform actions on computers in other zones or giving them access to other Active Directory objects.

How Role-Based Access Rights Can be Used

Role-based access rights are more flexible than Active Directory group membership because Active Directory groups provide static permissions. For example, if Jonah is a member the Active Directory Backup Operators group, he has all of the permissions defined for members of that group regardless of when or where he logs on to computers in the forest. In contrast, role assignments can be scheduled to start and end, apply only during specific hours, or only be available on specific computers. For example, Jonah may only be in the Backup Operators role on a specific computer or only on weekends.

Role-based access rights also prevent password sharing for privileged accounts, helping to ensure accountability. Users who need to be able to launch applications with elevated privileges can log on with their regular account credentials but run the application using an appropriate role without being prompted to provide the administrative password. For example, if Angela is assigned a role that enables her to run Disk Defragmenter using elevated privileges, she can log on with her normal credentials and select the role that enables her to run Disk Defragmenter without being prompted to provide an administrator user name and password.

Auditing User Activity on Windows Computers

Just as it is important to protect assets and resources from unauthorized access, it is equally important to track what users who have permission to access

those resources have done. For users who have privileged access to computers and applications with sensitive information, auditing helps ensure accountability and improve regulatory compliance. With the Audit and Monitoring service, you can capture detailed information about user activity and all of the events that occurred while a user was logged on to an audited computer.

If you choose to enable audit and monitoring service on Windows computers, the Agent starts recording user activity when a user selects a role or logs on to a computer. The agent continues recording until the user logs out or the computer is locked because of inactivity. The user activity captured includes an audit trail of the actions a user has taken and a video record of the applications opened, any text that was entered, and the results that were displayed on the screen. Because information about user activity, called a **session**, is collected as it happens, you can monitor computers for suspicious activity or troubleshoot problems immediately after they occur.

When users start a new session on an audited computer, they can be notified that their session is being audited and they cannot turn off auditing except by logging off. The information recorded is then transferred to a Microsoft SQL Server database so that it is available for querying and playback. You can search the stored user sessions to look for policy violations, user errors, or malicious activity that may have led to a service degradation or outage.

In addition to saving video record of user activity, sessions provide a summary of actions taken so that you can scan for potentially interesting or damaging actions without playing back a complete session. After you select a session of interest in the Audit Analyzer, the console displays an indexed list of actions taken in the order in which they occurred. You can then select any entry in the list to start viewing the session beginning with that action. For example, if a user opened an application that stores credit card information, you can scan the list of actions for the launch of that application and begin reviewing what happened in the session from that time until the user closed that application.

If users change their account permissions to take any action with elevated privileges, the change is recorded as an audit trail event. You can search for these events to find sessions of interest.

Using Access and Auditing Features Together

If you use the Access and Audit and Monitoring service features together, you can define role-based access rights, restrict when and where roles are available, identify roles that should be audited, trace activity when roles with elevated permissions are selected and used, and play back session activity based on the criteria you choose. However, audit and monitoring service requires database storage for the audited sessions and management of network communication for collecting and transferring audited sessions from computers being audited to one or more databases where the sessions are stored. You also need to decide which roles should require audit and monitoring service and the computers you want to audit.

This chapter provides an overview of the Delinea software architecture for identity management, privilege elevation, and auditing on Windows computers.

Identity and Privilege Management

In Server Suite, the authentication service and privilege elevation service provide role-based access control and privilege management for Windows computers. For administration, the services provide tools that help you define and manage access rights and roles for Active Directory users and groups. To enforce the rights and roles you define, you install an agent on each server or workstation to be managed.

Defining Rights and Roles Using Access Manager

When you install Server Suite, you choose the components you want to enable. For identity and privilege management, the key component for administration is the Access Manager console. Although there are other ways to define and manage access rights, roles, and role assignments, Access Manager is the primary tool for managing all of the Delinea software information stored in Active Directory. With Access Manager, you can:

- Create and manage zones to control access to all of the computers you support, including Windows, UNIX, Linux, and Mac OS X computers.
- Set and modify specific types of access right for users and groups.
- Add and customize the role definitions available in different zones, including any time restrictions on when roles are available or cannot be used.
- Assign and manage roles for individual Active Directory user or Active Directory groups.
- Associate groups of computers that share a common function or attribute with users who have a specific role assignment.
- Generate and view reports describing the users, groups, computers, and applications you are managing and which users and groups have access to which computers.
- View and manage licenses for servers and workstations.

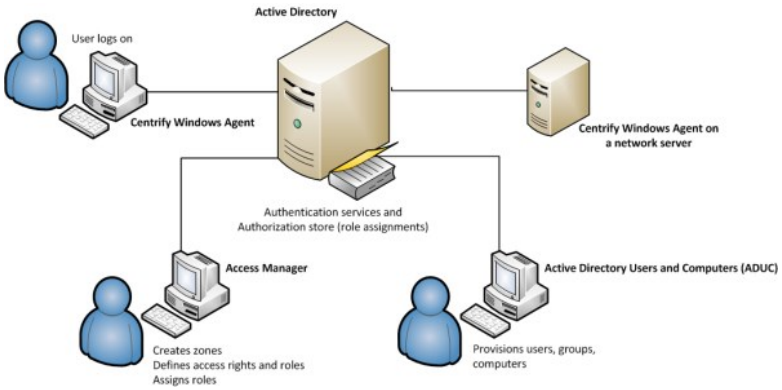
Enforcement of Rights and Roles by the Agent

For identity and privilege management, the key component for deployment is the Agent for Windows. After you install the agent on a server or workstation and identify a zone for the computer to join, the computer becomes a **Delinea-managed computer**. If you have enabled access management features for the agent, you can then define access rights and role-based policies to control what different sets of users can do on those computers in each zone.

After you deploy the Agent for Windows and select access management on a computer, the agent provides the following identity and privilege management features:

- Users logging on to the computer must be assigned to a role that allows them to log on.
- Users who are assigned to a role with **application rights** can run a specific application with elevated privileges.
- Users who are assigned to a role with **desktop rights** can create new Windows desktops that enables them to run all local applications with elevated privileges.
- Users who are assigned to a role with **network access rights** can connect to network resources with elevated privileges.

The following illustration provides a simplified view of the components for identity and privilege management.



In this illustration, an Agent is installed on an individual user's workstation and on a server accessed remotely. The administrative consoles that you use to manage zones, access rights, role definitions, and Active Directory accounts are installed on two separate computers. As shown in the illustration, all of these computers are part of an Active Directory domain and have access to an Active Directory domain controller. If you work with other platforms, the architecture

is the same but you would have additional platform-specific agents.

To ensure that you can centrally manage access to Windows computers with the privilege elevation service and the Agent for Windows, you should check that your network meets a few basic requirements:

- You have at least one Active Directory forest and domain controller.
- All of the computers you want to manage must be joined to an Active Directory domain and can communicate with an Active Directory domain controller over the network or through a firewall.
- You have a basic deployment plan in place that identifies your primary goals, team members and responsibilities, and a target set of computers.

The Audit and Monitoring Service Infrastructure

The Audit and Monitoring Service is part of Server Suite. The service captures detailed information about user activity on the computers you choose to audit.

Auditing Captures User Activity

After you deploy audit and monitoring service, the Agent for Windows captures all of the user activity on the computers you choose to audit. Depending on whether you enable identity and privilege management together with audit and monitoring service, or just audit and monitoring service on a computer, the agent starts recording user activity when a user selects a role or logs on to a computer and continues recording until the user logs out or the computer is locked because of inactivity. If you enable identity and privilege management together with audit and monitoring service on a computer, the agent records user activity when a role withaudit and monitoring service is used. If you only enable audit and monitoring service on a computer, all user activity is captured by default.

Each record of continuous user activity is called a **session**, and starts as soon as users log on, whether they log on locally, using a Windows Remote Desktop connection, through a virtual network connection such as Citrix or VNC, or using any other type of remote access software. A session ends when the user logs out, disconnects, or is inactive long enough to lock the desktop. If the user reconnects to a disconnected desktop or unlocks the desktop, the agent resumes recording the user's activity as a new session. Each session is a video record of everything that takes place on the user's desktop during a period of user activity.

Auditing Requires a Scalable Architecture

To ensure scalability for large organizations and fault tolerance, audit and monitoring service has a multi-tier architecture that consists of the following layers:

- **Audited computers** are the computers on which you want to monitor activity. To be audited, the computer must have an agent installed, audit features enabled, and be joined to an Active Directory domain.
- **Collectors** are intermediate services that receive and compress the captured activity from the agents on audited computers as it occurs. You should establish at least two collectors to ensure that audit and monitoring service is not interrupted. You can add collectors to your installation at any time, and it is common to have multiple collectors to provide load balancing and redundancy.
- **Audit stores** define a scope for audit and monitoring service and include the audit store databases that receive captured activity and audit trail records from the collectors and store it for querying and playback. Audit store databases also keep track of all the agents and collectors you deploy. For scalability and network efficiency, you can have multiple audit stores each with multiple databases.
- A **management database server** is a computer that hosts the Microsoft SQL Server instance with the audit management database. The management database stores information about the overall installation, such as the scope of each audit store, which audit store database is active, where there are attached databases, the audit roles you create, and the permissions you define. The management database enables centralized monitoring and reporting across all audit stores, collectors, and audited computers.
- **Audit Manager** and **Audit Analyzer consoles** are the graphical user interfaces which administrators can use to configure and manage the deployment of audit components, such as agents and collectors, or query and review captured user sessions.
- A **reporting database** collects data from audit stores and the management database and saves the data in a format that is optimized for reporting. With the reporting database, you can generate event notifications, such as when an audited system goes offline.

To ensure that audit data transferred over the network is secure, communication between components is authenticated and encrypted.

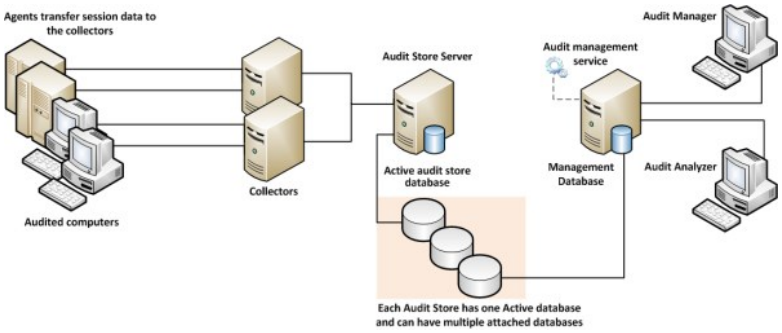
In addition to these core components of the audit and monitoring service infrastructure, there is a separate Windows service that is optional to collect audit trail events when there are audit store databases that are not accessible, for example, because of network issues or the database server is shut down. This audit management service spools the events on the management database, then sends them to the audit store database when the inaccessible database comes back online.

How Audited Sessions are Collected and Stored

The agent on each audited computer captures user activity and forwards it to a collector on a Windows computer. If the agent cannot connect to a collector—for example, because all of the computers hosting the collector service for the agent are shut down for maintenance—the agent spools the session data locally and transfers it to a collector later. The collector sends the data to an audit store server, where the audit data is stored in the Microsoft SQL Server database that you have designated as the **active audit store**. As you accumulate data, you can add more SQL Server databases to the audit store to hold historical information or to change the database designated as the active audit store database.

When an administrator or auditor uses the Audit Analyzer console to request session data, the audit management server retrieves it from the appropriate audit store.

The following figure illustrates the basic architecture and flow of data with a minimum number of audit and monitoring service components installed.



In the illustration, each agent connects to one collector. In a production environment, you can configure agents to allow connections to additional collectors for redundancy and load balancing or to prevent connections between specific agents and collectors. You can also add audit stores and configure which connections are allowed or restricted. The size and complexity of the auditing infrastructure depends on how you want to optimize your network topology, how many computers you are audit and monitoring service, how much audit data you want to collect and store, and how long you plan to retain audit records.

Deploying the Audit and Monitoring Service Infrastructure

The multi-tiered architecture of audit and monitoring service requires that you deploy an audit and monitoring service infrastructure to transfer and store the information captured by agents on the audited computers. This auditing infrastructure is referred to collectively as an **Auditing installation**. The audit and monitoring service installation represents a logical boundary similar to an Active Directory forest or site. It encompasses all of the audit and monitoring service components you have installed—agents, collectors, audit stores, management database, and consoles—regardless of how they are distributed on your network. The installation also defines the scope of audit data available. All queries and reports are against the audit data contained within the installation boundary.

The most common deployment scenario is to have a single audit and monitoring service installation for an entire organization so that all audit data and management of the audit data is centralized. Within a single audit and monitoring service installation, you can have components wherever they are needed, as long as you have the appropriate network connections that allow them to communicate with each other. The audit data for the entire installation is available to users who have permission to query and view it using a console. For most organizations, having a single audit and monitoring service installation is a scalable solution that allows a "separation of duties" security model through the use of audit roles. If you establish a single audit and monitoring service installation, there will be one Master Auditor role for the entire organization, and that Master Auditor can control the audit data that others users and groups can see or respond to by defining roles that limit access rights and privileges.

However, if you have different lines of business with different audit policies, in different geographic locations, or with different administrative groups, you can configure them as separate audit and monitoring service installations. For example, if you have offices in North America and Hong Kong managed by two different IT teams—IT-US and IT-HK—you might want to create two installations to maintain your existing separation of duties for the ITUS and IT-HK teams.

Planning Where to Install Audit and Monitoring Service Components

Before you install audit and monitoring service components, you should develop a basic deployment plan for how you will distribute and manage the components that make up an installation. For example, you should decide how many collectors and audit stores to create and where to put them. You should also consider the network connections required and how many computers you plan to audit. For example, you can have multiple agents using the same set of collectors, but you should keep the collectors within one hop of the agents they serve and within one hop of the audit stores to which they transfer data.

By planning where to install components initially, you can determine the number of collectors you should have for load balancing or redundancy. After the initial deployment, you can add collectors and audit stores whenever and wherever they are needed.

Using Multiple Databases in an Audit Store

Each audit store uses Microsoft SQL Server to provide database services to the installation. When you configure the first audit store, you identify the database instance to use for audit and monitoring service and that database becomes the active database for storing incoming audit data. A single audit store, however, can have several databases attached to it. Attached databases store historical information and respond to queries from the management database. You can use the Audit Manager console to control the databases that are attached and to designate which database is active. Only one database can be active in an audit store at any given time.

Although the audit store can use multiple databases, the presentation of session data is not affected. If a session spans two or more databases that are attached to the audit store, the Audit Analyzer console presents the data as a single, unbroken session. For example, if you change the active database during a session, some of the session data is stored in the attached database that is no longer active and some of it stored in the newly activated database, but the session data plays back as a single session to the auditor.

Using Multiple Consoles In an Installation

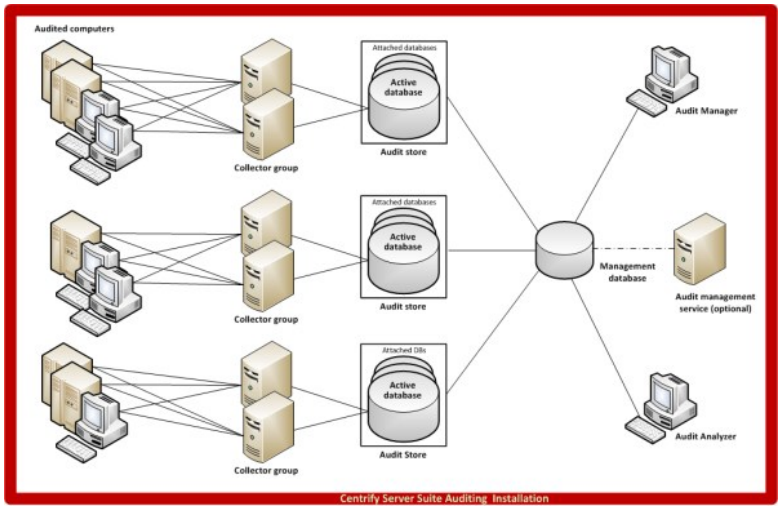
A single audit installation always has a single audit management server and database. In most cases, however, you use more than one console to request data from the audit management database. The two most important consoles in an installation are the Audit Manager console and the Audit Analyzer console.

- As an installation owner, you use the Audit Manager console to configure and manage the audit installation. In most organizations, there is only one Audit Manager console installed.
- Auditors and administrators use the Audit Analyzer console to search, retrieve, play back, and delete sessions. The auditor can use predefined queries to find sessions or define new queries. Auditors can also choose whether to share their queries with other auditors or keep them private. In most organizations, there are multiple Audit Analyzer consoles installed.

In addition to the Audit Manager and Audit Analyzer consoles, audit and monitoring service includes a settings control panel and a collector control panel.

- As an administrator, you can use the Audit & Monitoring Service Settings control panel to configure the agent on Windows. Normal users who log on and run applications on the audited computer cannot stop, pause, restart, or configure the agent.
- You can use the collector control panel to configure a collector.

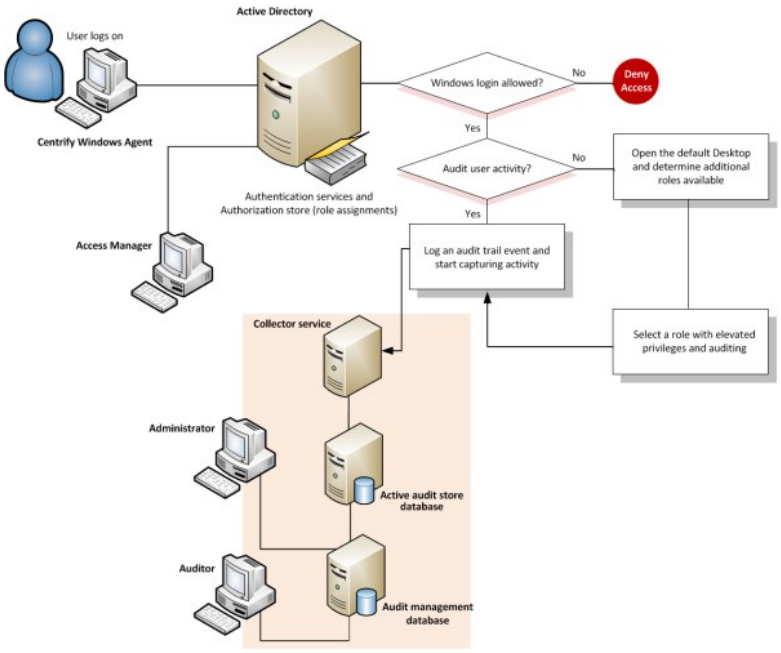
The following illustration is an example of the architecture of a medium-size installation.



Basic Operation with Identity and Privilege Management, and Auditing

When you combine identity and privilege management together with auditing on the same computer, you have an audit trail and video record of actions performed with elevated privileges. For example, when you deploy identity and privilege management features, users must be assigned to a role with permission to log on. If they are allowed to log on and audit and monitoring service is deployed, the agent begins auditing their activity. If a user creates a new desktop, opens a protected application, or connects to services on a remote network server with administrative or service account privileges, the action is recorded and can be traced back to the account used to log on.

The following illustration provides a simplified view of the architecture and flow of data when you deploy components for identity management, privilege management, and auditing.



Although it is not depicted in the illustration, the audit trail records every successful or failed attempt to use a role, including the login role. You do not have to enable audit and monitoring service for a role to record this information. Every computer that has the Agent for Windows records the use of elevated privileges by default. If you do enable audit and monitoring service for a role, however, you can record all of the user activity after the user switches to the audited role. With audit and monitoring service enabled, the audit trail and the user activity are stored in the database and available for display and analysis anywhere you install the Audit Analyzer console. Without audit and monitoring service, the audit trail is only available in the Windows event log on the local computer where the activity took place.

Planning a Deployment

This chapter describes the decisions you need to make during the planning phase of a deployment and summarizes what's involved in deploying identity management, privilege management, audit and monitoring service, and Agents. It includes simplified diagrams that highlight the steps involved.

Because of its multi-tier architecture and storage requirements, most of the information in this chapter applies to planning a deployment of audit and monitoring service. If you are only interested in deploying identity and privilege management without auditing, you should scan What's involved in the deployment process for relevant topics and continue to Installing Server Suite and updating Active Directory.

Why Planning is Important

Deploying Delinea software on Windows affects how users access local applications and remote services. These changes will become a critical part of your IT infrastructure and the management of your organization's resources. Therefore, it is important that you plan and test your deployment strategy and validate the results before placing Delinea software components into a production environment.

After you deploy Delinea software in a production environment, the rights and roles you define will control whether users can log on and what they can do on specific computers if they are allowed to log on. Because preventing users from accessing critical resources or services can affect business operations, you should analyze the requirements of your environment as thoroughly as possible before moving from a pilot deployment into production.

Identify Identity, Privilege Management, and Auditing Goals

As discussed in Managing Windows computers using Delinea software, you have the option of focusing your deployment on identity and privilege management, or on audit and monitoring service, or on a combination of the two. If you plan to install components for identity and privilege management together with audit and monitoring service, you can use roles and role assignments to control which users and groups are audited and under what circumstances auditing takes place. You can also capture detailed information about what happened after a user selected a role with domain administrator privileges or started an application using a service account.

During the planning phase, you should decide on the goals of your deployment—identity and privilege management, audit and monitoring service, or both—because that decision affects all of the other decisions you need to make. If you plan to include audit and monitoring service, you should also start to identify who and what you want to audit, any roles where no auditing should be done, and any roles that will require auditing.

Decide on the Scope of the Installation

Before you deploy any of the audit and monitoring service infrastructure, you should decide on the scope of the installation and whether you want to use a single installation for your entire Active Directory site, or separate installations for different geographical areas or functional groups.

The most common deployment is a single audit and monitoring service installation for each Active Directory forest, so that auditors can query and review information for the entire organization. However, if your Active Directory site has more than one forest, you might want to use more than one audit and monitoring service installation. If you want to use more than one audit and monitoring service installation, you should determine the subnetwork segments that will define the scope of each installation.

In Active Directory, a site represents the collection of Internet Protocol (IP) addresses that describe the physical structure of your network. If you are not familiar with how Active Directory sites are defined, you should consult Microsoft documentation for more information.

Decide Where to Install the Management Database

Each installation has a single audit management server and database. The management database is a Microsoft SQL Server database that stores information about the installation such as the Active Directory sites or subnets associated with each audit store.

The computer you use for the audit management database should have reliable, high-speed network connectivity. The management database does not store the captured sessions, and is, therefore, much smaller than the audit store databases. There are no specific sizing requirements or recommendations for the management database.

You can use the following guideline as the recommended hardware configuration for the computer you use as the management database:

Management database	Any	1 to 2	2.33 GHz	8 GB

Decide Where to Install Collectors and Audit Stores

Although a collector and an audit store database can be installed on the same computer for evaluation, you should avoid doing so in a production environment. As part of the planning process, therefore, you need to decide where to install collectors and audit store databases. In designing the network topology for the audit and monitoring service installation, there are several factors to consider. For example, you should consider the following:

- Database load and capacity
- Network connectivity
- Port requirements
- Active Directory requirements

The next sections provide guidelines and recommendations to help you decide where to install the collectors and audit store databases required to support the number of computers you plan to audit.

Use Separate Computers for Collectors and Audit Store Databases

To avoid overloading the computers that host collectors and audit store databases, you should install collectors and audit store SQL Server databases on separate computers. Because SQL Server uses physical memory to store database information for fast query results, you should use a dedicated computer for the audit store database, and allocate up to 80% of the computer's memory to SQL Server. In most installations, you also need to plan for more than one audit store database and to periodically rotate from one database to another to prevent any one database from getting too large. For more information about managing audit store databases, see Managing audit store databases.

Plan for Network Traffic and Data Storage

You should minimize the distance network packets have to travel between an agent and its collector. You should also minimize the distance between collectors and their audit stores. If possible, you should not have more than one gateway or router hop between an agent and its collector.

Default Ports for Network Traffic and Communication

To help you plan for network traffic, the following provides an overview of the network communications and ports used when a user logs on and the ports used in the initial set of network transactions.

When a user logs on, the Agent for Windows connects to Active Directory to begin the lookup process, then the agent and the domain controller exchange messages as follows:

- Directory Service - Global Catalog lookup request on port 3268.
- Authentication Services - LDAP sealed request on port 389.
- Kerberos – Ticket Granting Ticket (TGT) request on port 88.
- Network Time Protocol (NTP) Server – Time synchronized for Kerberos on port 123.
- Domain Name Service (DNS) – Host (A), Pointer (PTR), Service Location (SRV) records on port 53.
- RPC over TCP - For inbound RPC endpoint mapper connections to support network discovery or if password management and validation uses RPC over TCP on port 135.

Depending on the specific components you deploy and operations performed, you might need to open additional ports. The following table summarizes the ports used for different editions of Delinea software.

22	Encrypted TCP communication for OpenSSH connections	Authentication service and privilege elevation service for secure shell connections on remote clients.
23	TCP communication for Telnet connections	Delinea authentication service, privilege elevation service, and audit and monitoring service. By default, telnet connections are not allowed because passwords are transferred over the network as plain text.
53	TCP/UDP communication	Authentication service and privilege elevation service, clients use the Active Directory DNS server for DNS lookup requests.

88	Encrypted UDP communication	Authentication service and privilege elevation service, Kerberos ticket validation and authentication, agents, Delinea PuTTY
123	UDP communication for simple network time protocol (NTP)	Authentication service and privilege elevation service, keeps time synchronized between clients and Active Directory for Kerberos ticketing.
389	Encrypted TCP/UDP communication	authentication service and privilege elevation service, Active Directory authentication and client LDAP service.
443	Cloud proxy server to Delinea cloud service	Delinea software for mobile device management.
445	Encrypted TCP/UDP communication for delivery of group policies	Authentication service and privilege elevation service, adclient and adgpupdate use Samba (SMB) and Windows file sharing to download and update group policies, if applicable.
464	Encrypted TCP/UDP communication for Kerberos password changes	Authentication service and privilege elevation service, Kerberos ticket validation and authentication for agents,Delinea PuTTY, adpasswd, and passwd.
1433	Encrypted TCP communication for the collector connection to Microsoft SQL Server	Authentication service, privilege elevation service, and audit and monitoring service; collector service sends audited activity to the database.
3268	Encrypted TCP communication	Authentication service and privilege elevation service, Active Directory authentication and LDAP global catalog updates.
5063	Encrypted TCP/RPC communication for the agent connection to collectors	Authentication service, privilege elevation service, and audit and monitoring service; auditing service records user activity on an audited computer.
none	ICMP (ping) connections	Authentication service and privilege elevation service, to determine whether if a remote computer is reachable.

Auditing Requires Database Management

If you are planning a deployment with just audit and monitoring service or with identity management, privilege management, and auditing, you must plan how you will create and manage the databases that receive and store audit data. You should also consider your data archiving and retention policies, who should be given auditor permissions, and other details because these decisions affect your storage and maintenance requirements. For more information about managing an installation for auditing, see Managing auditing for an installation.

For audit and monitoring service, you should plan a pilot deployment of 20 to 25 agents to determine how much audit data your organization would generate and how fast the database can increase in size as you add agents. For more information about monitoring a pilot deployment for audit and monitoring service and guidelines for sizing the database, see Estimating database requirements based on the data you collect.

Identify an Active Directory Site or Subnets

Depending on the size and distribution of your Active Directory site, an audit store might cover an entire site or specific subnet segments. If you have a large, widely distributed site, you should consider network connectivity and latency issues in determining which subnets each audit store should serve. In addition, you should always place collectors in the same site as the agents from which they receive data. Collectors and agents must always be in the same Active Directory forest. If possible, you should put collectors and agents in the same domain.

> **Note**: If you deploy agents in a perimeter network, such as a demilitarized zone (DMZ), that is separated from your main network by a firewall, put the collectors in the same Active Directory domain as the audited computers. The collectors can communicate with the audit store database through a firewall.

Determine How Many Collectors and Audit Stores to Install

Although you can add collectors and audit stores to your audit and monitoring service installation after the initial deployment, you might want to calculate how many you will need before you begin deploying components. You should always have at least two collectors to provide redundancy. As you increase the number of agents deployed, you should consider adding collectors.

Estimate the Number of Agents and Sessions Audited

If you plan to use more than the minimum number of collectors, the most important factor to consider is the number of concurrent sessions you expect to monitor on audited computers. The number of concurrent sessions represents the number of interactive users that the agent is actively capturing for at the same time.

You can use the following guidelines as a starting point and adjust after you have observed how much audit data you are collecting and storing for Windows computers:

up to 100 agents	2	1
more than 100 agents	2 for every 100 agents	1 for every 100 agents

Determine the Recommended Hardware Configuration

The hardware requirements for collectors and audit store servers depend on the size of the installation and where the components are installed on the network. For example, the requirements for a computer that hosts the collector service are determined by the number of audited computers the collector supports, the level of user activity being captured and transferred, and the speed of the network connection between the agents and the collector and between the collector and its audit store.

You can use the following guidelines as the recommended hardware configuration for the computers you use as collectors and audit store servers when auditing Windows computers:

Collectors	Up to 100 active agents	2	2.33 GHz	8 GB
Audit store	Up to 200 active agents	2	2.33 GHz	8 GB
	200 to 500 active agent	4	2.33 GHz	32 GB

Guidelines for Storage

Because audit and monitoring service collectors send captured user sessions to the active SQL Server database, you should optimize SQL Server storage for fast data logging, if possible. For the active database, you get the most benefit from improvements to disk write performance. Read performance is secondary. Fibre Attached Storage (FAS) and Storage Area Network (SAN) solutions can provide 2 to 10 times better performance than Direct Attached Storage (DAS), but at a higher cost. For attached databases that are only used to store information for queries, you can use lower cost storage options.

Guidelines for Disk Layout

The following table outlines the recommended disk arrays:

Operating system	C: RAID 1	Operating system files, page file, and SQL Server binaries.
Microsoft SQL Server	D: RAID 10 (1+0)	Audit store database.
	E: RAID 10 (1+0)	Audit database log files.
	F: RAID 1 or 10 (1+0)	Temporary database space (tempdb) for large queries for reports.

G: RAID 1	Database dump files.	

The size of disk needed depends on the number, length, and types of sessions recorded each day, the selected recovery model, and your data retention policies. For more information about managing audit store databases, see Managing audit store databases.

Decide Where to Install Agents

The Agent for Windows must be installed on all of the computers you want to audit. Therefore, as part of your planning process, you should decide whether you want to audit every computer on the network or specific computers, such as the computers used as servers or used to run administrative software.

Before installing the agent, verify the following:

- The computer is joined to Active Directory.
- The computer has Windows security update KB3033929 installed if it is running Windows 7 with Service Pack 1 or Windows Server 2008 R2 with Service Pack 1.
- The computer has .NET 4.6.2 or later installed.
- The computer has Windows Installer version 4.5 or newer.
- Agents can communicate with a collector only if the agents and collector are in the same Active Directory forest.

Decide Where to Install Consoles

You can install and run the Audit Manager console and the Audit Analyzer console on the same computer or on different computers. The computers where you install the consoles must be joined to the Active Directory domain and be able to access the management server and the database that serves the installation.

You can also use the Audit Analyzer console to run queries from any additional computers with network access to the management database. Therefore, you should decide where it would be convenient to have this capability.

Check SQL Server Logins for Auditing

An audit installation requires at least two Microsoft SQL Server databases: one for the management database and at least one for the first audit store database. To successfully connect to these databases, you must ensure that the appropriate users and computers have permission to read or to read and write for the databases that store audit-related information.

The simplest way to manage SQL logins for auditors and administrators is to do the following:

- Ensure you have a SQL login account for the NT Authority\System built-in account.
- Add the NT Authority\System account to the system administrator role.
- Use Audit Manager to grant Manage SQL Logins permissions to the Active Directory users and groups that require them.

If you use Audit Manager to manage SQL logins, you can use Active Directory membership to automatically add and remove the permissions required for auditing activity. There is no requirement to use the SQL Server Management Studio to manage logins or permissions. Because it is recommended that you have a dedicated SQL Server instance for auditing, giving the NT Authority\System account a SQL login and system administrator role is an acceptable solution for most organizations.

Create Security Groups for Auditing

Depending on whether you configure Microsoft SQL Server to use Windows only authentication or Windows or SQL Server authentication, your SQL Server login credentials might be a Windows account or a SQL Server login account that is not associated with a Windows account.

To facilitate communication and the management of SQL logins, you can create Active Directory security groups for the following users and computers:

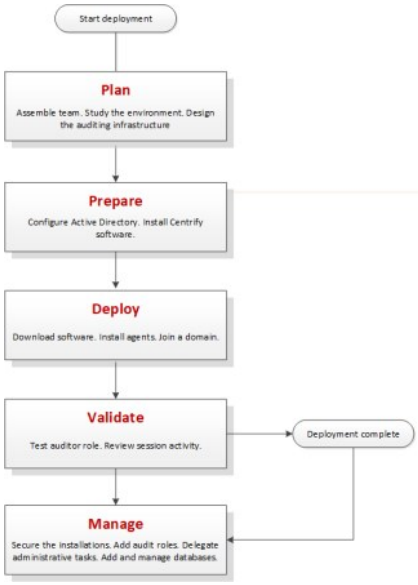
- **Admins** for the user accounts that perform administrative tasks using Audit Manager.
- **Auditors** for the user accounts that user Audit Analyzer.
- **TrustedCollectors** for the computers accounts that host the collector service.

If you create these Active Directory security groups, you can then use Audit Manager to grant Manage SQL Login permissions for each group to allow its members to connect to the appropriate SQL Server database. Creating Active Directory security groups with SQL Server logins enables you to manage access to the databases required for auditing through Active Directory group membership without the help of the database administrator.

Any time you want to add an administrator, auditor, or collector computer to the installation, you simply add that user account or computer object to the appropriate Active Directory group. If an administrator or auditor leaves or if you want to stop using the collector on a particular computer, you can remove that user or computer from its Active Directory security group to prevent it from accessing the database.

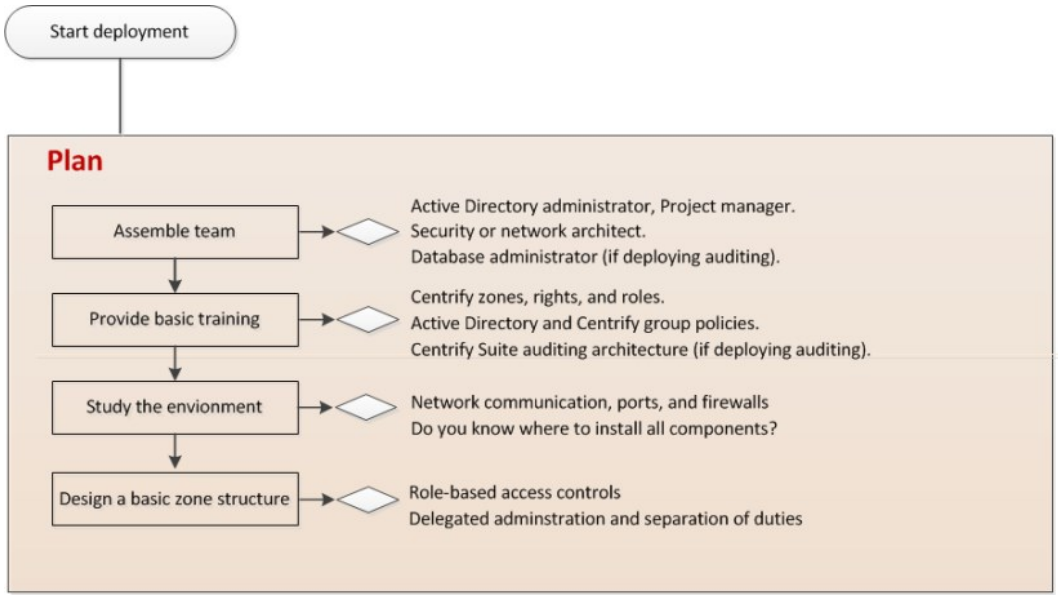
What's Involved in the Deployment Process

Most of the planning in this chapter has focused on designing the audit and monitoring service infrastructure and deciding where to install components. The following illustration provides a visual summary of the complete deployment process and highlights the keys to success. The sections after the flowchart provide additional details about what's involved in each phase or the decisions you will need to make, such as who should be part of the deployment team, where to install the software, and who has permission to do what.



Plan

During the first phase of the deployment, you collect and analyze details about your organization's requirements and goals. You can then also make preliminary decisions about sizing, network communication, where to install components, and what your zone structure should look like.



Here are the key steps involved:

- Identify the goals of the deployment.

  - Is identity and privilege management or audit and monitoring service a primary goal?
  - Are identity and privilege management and audit and monitoring service equally important to the organization?
  - Is audit and monitoring service important for specific computers?
  - Is audit and monitoring service important for computers used to perform administrative tasks?
  - Is audit and monitoring service important for computers that host specific applications or sensitive information?
  - Should audit and monitoring service be required for users in specific groups or with specific roles?

    For example, if audit and monitoring service is important, are you primarily interested in auditing Windows servers, such as SQL Server, Exchange, and IIS, administrative workstations, or computers that host specific applications or sensitive information?

- Assemble a deployment team with Active Directory and other expertise.
  - People with specific knowledge, such as Exchange, IIS, or Sharepoint administrators.
  - If auditing, at least one Microsoft SQL Server database administrator.
- Provide basic training on Delinea software architecture, concepts, and terminology.
- Study the existing environment to identify target computers where you plan to install Delinea software components.
  - Plan for permissions and the appropriate separation of duties for your organization.
  - Review network connections, port requirements, firewall configuration.

    For more information about network communication and the ports used, see Plan for network traffic and data storage.

  - Identify computers for administration.
    - **Basic deployment** — Access Manager
    - **Auditing** — Audit Manager and Audit Analyzer consoles
  - Identify computers to be used as collectors, audit stores, and the management database.
    - Verify that you have reliable, high-speed network connections between components that collect and transfer audit data.
    - Verify you have sufficient disk storage for the first audit store database.
  - Identify the initial target group of computers to be managed and audited.
- Design a basic zone structure that suits your organization.
  - Single or multiple top-level parents.
  - Initial child zones, for example, separate zones for different functional departments or administrative groups.

Prepare

After you have analyzed the environment, you should prepare the Active Directory organizational units and groups to use. You can then install administrative consoles and the audit and monitoring service infrastructure, and prepare initial zones.

Here are the key steps involved:

- (Optional) Create organizational units or containers to define a scope of authority.

  The deployment team should consult with the Active Directory enterprise administrator to determine whether any additional containers or organizational units would be useful, who should be responsible for creating Licenses and Zones container objects, and who will manage the objects in those containers.

- (Optional) Create the additional Active Directory security groups for your organization.

  Groups can simplify permission management and the separation of duties.

- Install Access Manager on at least one administrative Windows computer.
- Open Access Manager for the first time to run the Setup Wizard for the Active Directory domain.
- Create a parent zone and the appropriate child zones as identified in your basic zone design.

  The hierarchical zone structure you use depends primarily on how you want to use inheritance and roles.
- Prepare Windows computer accounts in the appropriate zones and assign the default Windows Login role to the appropriate Active Directory users and groups.
- Install Audit Manager and Audit Analyzer together or separately.
- Create an installation and a management database on one computer.
- Create an audit store and audit store database on at least one computer.
- Install a collector on at least two computers.

Deploy

After you have prepared Active Directory, installed administrative consoles on at least one computer, created at least one zone, and prepared the audit and monitoring service infrastructure, you are ready to deploy on the computers to be managed.

Here are the key steps involved:

- Create Desktop, Application, and Network Access rights.
- Add Desktop, Application, and Network Access rights to custom role definitions.
- Assign custom roles to the appropriate Active Directory users and groups.
- Install the Agent for Windows on a target set of computers.
- Join the appropriate zones.
- Prepare a Group Policy Object for deploying agents remotely using a group policy.
- Assign the appropriate permissions to the users and groups who should have access to audit data.

Validate

After you have deployed agents on target computers, you should test and verify operations before deploying on additional computers.

Here are the key steps involved:

- Log on locally to a target computer using an Active Directory user account and password to verify Active Directory authentication and Windows Login role assignment.
- Open a Remote Desktop Connection to a target computer to verify Active Directory authentication and Windows Login role assignment on a remote computer.
- Create a new desktop that gives you administrative rights and verify that you can start and stop Windows services or perform other administrative tasks.
- Right-click an application, select Run using selected roles, then select an available role for running the application.
- Open Audit Analyzer and query for your user session if audit and monitoring service is enabled.

Manage

After you have tested and verified identity management, privilege management, and audit and monitoring service operations, you are ready to begin managing the installation and refining on-going operations.

Here are the key steps involved if you deploying identity management, privilege management, and auditing for Windows computers:

- Secure the installation.
- Add roles and assign roles and permissions to the appropriate users, groups, and computers.
- Delegate administrative tasks to the appropriate users and groups for each zone.
- Deploy additional group policies on the appropriate organizational units.
- Create new databases and rotate the active database.
- Archive and delete old audit data.

- Automate key administrative tasks using Delinea-defined Powershell-based cmdlets and scripts.

Delinea

Authentication and Privilege Elevation Services Deployment Checklist

The following checklist provides an overview of each of the main steps that are involved when you deploy the Authentication Service and Privilege Elevation Service. For any tasks related to Delinea software, there are links to more information and procedures.

For auditing deployment steps, please see the Audit & Monitoring Service deployment checklist.

	PREPARATION AND PLANNING		
1	Analyze your network topology to determine where to install components and services and any hardware or software updates required.		Planning a Deployment
2	Create a list of the computers where you plan to install different components.		Planning a Deployment
3	Determine how you plan to install the software onto your computers.		Planning a Deployment
	PRE-INSTALL TASKS		
4	Prepare a domain account that has permissions to create Active Directory containers and child objects.	You'll need this account to create the OU using the Installation wizard.	
5	Prepare an Active Directory group to be zone administrators.		
6	Create the Zone Provisioning Agent (ZPA) service account.	Requires Active Directory domain admin privileges	
7	Apply group policy to allow the ZPA to run as a service.	Requires Active Directory domain admin privileges	
	INSTALL TASKS		
8	Install the Access Manager console, ZPA, group policies, create the OU in Active Directory, and so forth.		Installing Server Suite
9	(Optional) Configure ZPA – this is only needed if you plan on automatically provisioning users.		
10	Run adcheck on any UNIX computer that you want to manage and fix any issues until adcheck produces no issues.		
11	Install a Agent for Windows on each Windows computer that you want to manage.		Installing the Agent for Windows
12	Install a Agent for *NIX on each UNIX or Linux computer that you want to manage.		
13	Install additional Access Manager consoles on any Windows computer that you want to use for the Authentication and Privilege Management services.		Installing Additional Consoles
14	Verify that agents are working correctly. Run adinfo on managed UNIX computers.		Troubleshooting and Common Questions
	POST-INSTALL HOUSEKEEPING		

15	Identify UNIX users who do not have an Active Directory account.	Automatically done by adimport	adimport man page
16	Identify service accounts.		
17	Collect and analyze sudoers files.		
18	Create a list of roles in sudoers that will be migrated to Privilege Elevation Service.		
19	Create a list of users and groups to be migrated to Active Directory.		
20	Create missing Active Directory user accounts.		
	SETUP AND CONFIGURATION		
21	Create list of computers that will be joined to each zone.		
22	Create parent and child zones.		Creating a New Parent Zone Creating Child Zones
23	Delegate control to zones.		Delegating Control of Administrative Tasks
24	Import UNIX users and groups into Active Directory.		
25	Create Zone Provisioning groups and add users and groups to them.		
26	Pre-create computer objects in zones.		
27	Create role groups .		
28	Assign roles and users to role groups.		
29	Create ComputerRoles and ComputerRole groups.		Create a New Computer Role
30	Assign roles, users, and computers to ComputerRole groups.		Add Role Assignments to the Computer Role
31	Use “Show Effective Users” to check that profiles and roles are correct.		
32	Start the ZPA agent.	You configured ZPA in a previous step.	
33	Configure the ZPA provisioning rules for the parent zone.		
34	Join UNIX servers to Zones.		

35	Change the UID/GID of files for those users who have been assigned a new UID/GID in the Zone. Run adfixid on servers.	* Critical task that must be carefully coordinated with the users. Can be done at time of join to Active Directory with a script.	
	FINAL TASKS		
36	Check the status of the join and roles on the servers.	Run adflush, adinfo and dzinfo	
37	Back up passwd, shadow, and group files.		
38	Remove the users and groups (that have been migrated to Active Directory) from the local files.	Run adrmlocal on servers	

Accounts and Permissions for Installation and Deployment

Below is a summary of the account permissions that you need to install and deploy Server Suite.

Authentication and Privilege Elevation Services permissions

Access Manager Account Permissions

n/a	Domain administrator (when running Access Manager for the first time)	domain admin (in most cases)	Because the Setup Wizard creates container objects, you might need to use a domain administrator account. This requirement depends on the specific permissions your organization has configured for different classes of users. For example, if your organization only permits Domain Admins to create parent and child objects in Active Directory, you need to use an account with those permissions to run the Setup Wizard.

For more information, see:

- "Running Access Manager for the First Time" and "Permissions Required to use the Setup Wizard" in the Unexpected Link Text

Zone Provisioning Agent Account Permissions

Cfy_SVC_ZPA	Active Directory account	**Log on as a service**	The Zone Provisioning Agent requires permission to create UNIX profiles-- that is, the service connection points in each zone where it needs to perform provisioning operations. The service account that runs the Zone Provisioning Agent requires the **Log on as a service** right set as a local computer security policy, or in the default domain policy.

For more information, see:

- "About Zone Provisioning Agent and its Requirements" in the Unexpected Link Text

Report Services Account Permissions

report service account to run the Reporting Service	For domain-based reporting: Replicating directory changes at the domain level (ADUC) and replicate directory changes in ADSI For zone-based reporting: Read permission	Log on as a service		
SQL Server service account to run SQL Server	n/a	Log on as a service		member of the securityadmin role
PostgreSQL service				the account must have permission to connect to

account				PostgreSQL and create a database
report admin to run the Report Configuration wizard or the Upgrade & Deployment wizard and deploy reports to an existing SQL Server instance	needs to be a member of the domain	n/a	Folder Settings > Content Manager role	member of the securityadmin role (At the very least, the user needs permission to connect to SQL Server and create a database.)
report admin to modify the Reports Control Panel	Read permission to the domain root object of the selected domain. Read permission to all computer objects in the selected domain.	n/a		
Report viewer to view reports from SSRS/Internet Explorer			Site settings > System user role Folder settings > browser (assign SSRS roles to Active Directory group or users)	
Report writer read, write, edit access for reports, in addition to the permissions needed to view reports			Site settings > System user role Folder settings > Content Manager role (assign SSRS roles to Active Directory group or users)	

SQL Server Permissions Set by the Report Services Configuration Wizard

report services account to run the SQL Server Reporting Service	Snapshot Service (predefined role)
SQL Server service account to run SQL Server	If you deploy to an existing SQL Server instance, the configuration wizard makes no changes to the SQL Server service account. If you deploy to a new SQL Server instance: --If the operating system is Windows 2008 and you're using a SQL Server version later than 2012, virtual accounts are used for various SQL Server components, as follows: SQL Server engine: NT SERVICE\MSSQL$<InstanceName> SQL Server Agent: NT SERVICE\SQLAgent$<InstanceName> Full text search: NT SERVICE\MSSQLFDLauncher$<InstanceName>

	SSRS: NT SERVICE\ReportServer$<InstanceName> --Otherwise, the SQL Server service accounts are configured as follows: SQL Server engine: NT Authority\Network Service SQL Server Agent: NT Authority\Network Service Full text search: NT Authority\Local Service SSRS: NT Authority\Local Service
report admin to run the Report Configuration Wizard and deploy reports to an existing SQL Server instance	Connect SQL (cannot be revoked after setup) Create Database, Create any database, or Alter any database member of securityadmin role, or Alter any login permission
report admin to modify the Reports Control Panel	SnapshotAdmin (predefined role)
Report viewer to view reports from SSRS/Internet Explorer	Login permission SnapshotViewer (predefined role)
Report writer read, write, edit access for reports, in addition to the permissions needed to view reports	Login permission SnapshotViewer (predefined role)

> **Note**: Microsoft SQL Server Reporting System (SSRS) affords only role-based security in their reports. Be sure to grant appropriate access to reports. For example, if a user has access to only some data in the specified domain but all reports, they will be able to view all reports on all data from Active Directory.

For more information, see:

- "Required User Permissions for Report Services" and "SQL Server Permissions that are Set by the Configuration Wizard" in the Unexpected Link Text

Audit & Monitoring Permissions

Auditing permissions for SQL Server

NT Authority\System	machine account	SQL Server Roles: sysadmin role

Auditing security groups

Admins for the user accounts that perform administrative tasks using Audit Manager.	Active Directory	no explicit SQL Server permissions needed — Audit Manager handles the SQL Server permissions	Creating Active Directory security groups with SQL Server logins enables you to manage access to the databases required for auditing through Active Directory group membership without the help of the database administrator.
Auditors for the user accounts that use Audit Analyzer.			
Collectors for the computer accounts that host the			

collector service.

For more information, see Checking SQL Server Logins for Auditing.

This chapter describes how to install Server Suite software on Windows computers in a production environment. It includes instructions for installing all identity and privilege management, audit and monitoring service, and multi-factor authentication components. It also describes how to install the Agent for Windows, and how to enable services on agent-managed Windows computers.

If your deployment plan includes identity and privilege management, as well as audit and monitoring service, you should review the details in Planning a deployment before installing any components.

In a production environment, you should use separate computers for different components to ensure scalability and performance. For information about setting up an evaluation environment on a single computer for testing, see the *Evaluation Guide for Windows*.

Installation Checklist

As a preview of what's involved in the installation process, the following steps summarize what you need to do and the information you should have on hand for a successful deployment of Server Suite.

To prepare for installation:

1. Analyze your network topology to determine where to install components and services and any hardware or software updates required.

   For a review of the decisions to make and recommended hardware configuration, see Planning a deployment.

2. Create a list of the computers where you plan to install different components.

   For example, list the computers where you plan to install agents, collectors, audit store databases, consoles, and group policy extensions.

   If you are installing the audit and monitoring service infrastructure, you should use a dedicated computer for each component, so that the audit collector service, audit store database, and audit management database are on separate computers with high-speed and reliable network connectivity.

   For a review of the requirements associated with each component, see Planning a deployment.

3. Determine the scope of the audit installation.

   The most common deployment scenario is a single installation for an Active Directory site, but you can have more than one installation, if needed, and use subnets to limit the scope of the installation. If you are only implementing access management, you can skip this step, Step 4, and Step 7 through Step 10.

   For a review of what constitutes an installation, see Deploying the audit and monitoring service infrastructure and Decide on the scope of the installation.

4. Create Active Directory security groups for managing the permissions required for the audit and monitoring service infrastructure.

   For a review of the Active Directory security groups to create, see Create security groups for auditing. If you are only implementing identity and privilege management, you can skip this step.

5. Install Access Manager on at least one computer that can connect to the Active Directory forest.

6. Open Access Manager and add containers for licenses and zones to the Active Directory forest.

7. Install Microsoft SQL Server.

   If you are not a database administrator in your organization, you should submit a service request or contact an administrator who has permission to create databases for assistance. For more information about preparing a SQL Server database engine for auditing, see Installing and configuring Microsoft SQL Server for auditing. If you are only implementing access management, you can skip this step.

8. Install Audit Manager and Audit Analyzer.

   For more information about installing these products, see Installing the Audit Manager and Audit Analyzer consoles. If you are only implementing identity and privilege management, you can skip this step.

9. Open Audit Manager to create a new installation for auditing.

   For more information about using Audit Manager to create a new installation and audit store, see Creating a new installation. If you are only

implementing identity and privilege management, you can skip this step.

10. Install the audit collector service on at least two Windows computers.

    You can add collectors to the installation at any time. For more information about installing and configuring collectors, see Installing and configuring audit collectors. If you are only implementing identity and privilege management, you can skip this step.

11. Install an Agent for Windows on each Windows computer that you want to manage or audit.

    For more information about installing and configuring Agent for Windows, see Installing the Agent for Windows.

12. Install additional consoles on any Windows computer that you want to use for identity and privilege management, or audit and monitoring service.

After the initial deployment, you can add new agents, collectors, audit stores, and audit store databases to the audit installation or create additional installations at any time.

Installing Server Suite and Updating Active Directory

When you install Server Suite, components for the following features are installed:

- The Identity Platform, which enables MFA login, endpoints, and other platform services.
- The Privilege Elevation Service, which enables users and zone-joined computers to have elevated privileges.
- The Audit & Monitoring Service, which enables audit and monitoring service data to be collected and stored.
- The Agent for Windows, which enables each computer where the agent is installed to be managed by Server Suite software.
- The Licensing Service, which works together with Server Suite components to monitor and report usage and activity for all types of licenses. For more information about the licensing service, see the *License Management Administrator's Guide*

You can select which features to install from the Delinea software setup program.

After Server Suite are installed, you must enable some or all of them on each agent-managed computer. The enablement step lets you decide which services are available on each agent-managed computer.

Things to remember

- At least one zone must be created before an agent-managed computer can be enabled to use the identity and privilege management features that you install. If no zones are available, the agent-managed computer will not have the option of being joined to the authentication and privilege elevation services.
- When the Agent is upgraded or when it adds the Identity Platform, a corporate endpoint enrollment is performed in the Privileged Access Service. The endpoint device moves into the endpoint category and the device is marked as corporate owned.

Running the Setup Program on a Windows Computer

You can install components for all Server Suite from the Server SuiteCD or a downloaded ISO or ZIP file. After you access the distribution media, the setup or autorun program copies the necessary files to the local Windows computer. There are no special permissions required to run the setup or autorun program other than permission to install files on the local computer.

To install Delinea software on Windows:

1. Log on to the computer you have selected for administrative tasks and browse to the location where you have saved downloaded Delinea software files.

   If you have a physical CD, the Getting Started page is displayed automatically. If the page is not displayed, open the autorun.exe file to start the installation of Delinea software.

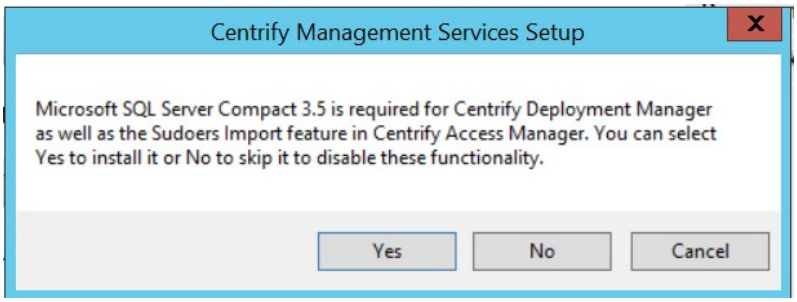
2. On the Getting Started page, click **Authentication & Privilege** to start the setup program for authentication and privilege elevation services.

   Note: {/b}The **Authentication & Privilege** components are the recommended first components to install so that Access Manager is available for you to use to create zones. At least one zone must be created before you can enable the authentication and privilege elevation services on an agent-managed computer.

   If any programs must be updated before installing, the setup program displays the updates required and allows you to install them. After updates are complete, you can restart the setup program.

3. At the following screen, select **Yes** to install Microsoft SQL Server Compact. The Access Manager console uses the Microsoft SQL Server Compact for

storage.



If you select No, Microsoft SQL Server Compact is not installed and some features of Access Manager are not available.

4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I agree to these terms**, then click **Next**.
6. Type your name and company name, then click **Next**.
7. Expand and select the Administration and Utilities components you want to install, then click **Next**.

   If you are only managing identity and privileges for Windows computers, you can install a subset of the components. For a Windows-only deployment, select the following components:

   - **ADUC property page extension** if you want to include profiles when displaying properties in Active Directory Users and Computers.
   - **Access Manager console (all)** if you want to use an administrative console to manage zones and roles.
   - **Group Policy Management Editor extension** if you want to deploy group policies.

     Installing Report Services is optional. If you select this option, see Installing and configuring Microsoft SQL Server for auditing for additional details.

     For a Windows-only deployment, you can deselect Utilities to skip the installation of those components.
8. Accept the default location for installing components, or click **Browse** to select a different location, then click **Next**.
9. Review the components you have selected, then click **Next**.

   The setup program begins installing the selected components.
10. Click **Finish** to complete installation.
11. Optionally install additional Server Suite components as follows:

    - **Licensing Service.** This service is installed by default when you install the Authentication & Privilege components, and usually does not need to be installed separately. For more information about the licensing service, see the *License Management Administrator's Guide*.
    - **Audit & Monitor.** The Auditing and Monitoring Service is not installed automatically with any other components, and must be installed separately if you intend to use auditing and monitoring features. For installation details, see Installing the Audit Manager and Audit Analyzer consoles.
    - **Agent for Windows.** To install the agent on client Windows computers so that those computers can be managed by Server Suite, see Installing the Agent for Windows.

Opening Access Manager to Update Active Directory

The first time you start Access Manager, a Setup Wizard prepares the Active Directory forest with parent containers for licenses and zones. The Setup Wizard also sets the appropriate permissions for the objects automatically. For more information about using the Setup Wizard to update Active Directory, see Starting Access Manager for the first time.

Installing and Configuring Microsoft SQL Server for Auditing

If you want to audit user activity on Windows, you must have at least one Microsoft SQL Server database instance for the audit management database and audit store databases. We recommend that you use a dedicated instance of SQL Server for the audit management database. A dedicated SQL Server instance is an instance that does not share resources with other applications. The audit store databases can use the same dedicated instance of SQL Server or their own dedicated instances.

There are three database deployment scenarios for your installation:

- **Evaluation**—Use the SQL Server Express with Advanced Services setup program (SQLEXPRADV_x64_ENU.exe) to create a new instance of Microsoft SQL Server Express. *You should only use Microsoft SQL Server Express for evaluation or for limited use in a test environment*. You should not use SQL Server Express databases in a production environment.

  If you choose to install a different version of Microsoft SQL Server Express for an evaluation and the version requires .NET version 3.5 SP1, you will need to manually install the .NET files yourself (the installer doesn't include these files).

- **Manual installation with system administrator privileges**—Install a Microsoft SQL Server database instance for which you are a system administrator or have been added to the system administrator role.

- **Manual installation without system administrator privileges**—Have the database administrator (DBA) install an instance of Microsoft SQL Server and provide you with system administrator credentials or information about the database instance so that you can create the management database and audit store databases.

Downloading and Installing SQL Server Manually

You can use an existing Microsoft SQL Server database engine or install a new instance. You can download Microsoft SQL Server software from the Microsoft website or through the Support Portal](https://www.delinea.com/login/?portal=support). In selecting a version of Microsoft SQL Server to download, you should be sure it includes Advanced Services. Advanced Services are required to support querying using SQL Server full-text search.

After downloading an appropriate software package, run the setup program using your Active Directory domain account and follow the prompts displayed to complete the installation of the SQL Server database engine.

Configuring SQL Server to Prepare for Audit and Monitoring Service

After you install the SQL Server database engine and management tools, you should configure the SQL Server instance for audit and monitoring service by doing the following:

- Depending on the version of SQL Server you install, you might need to manually enable full text search. For example, use SQL Server Surface Area Configuration for Services and Connections to start the full-text search service.
- Use SQL Server Configuration Manager to enable remote connections for TCP/IP.
- Use SQL Server Configuration Manager to restart the SQL Server and SQL Server Browser services.
- Verify whether SQL Server is using the default TCP port 1433 for network communications. If you use a different port, you should note the port number because you will need to specify in the server name when you create the management and audit store databases.

Installing the Audit Manager and Audit Analyzer Consoles

You can install Audit Manager and Audit Analyzer on the same computer or on different computers. The computers where you install the consoles must be joined to the Active Directory domain and be able to access the audit management database.

In most cases, the consoles are installed together on at least one computer.

To install Audit Manager and Audit Analyzer on the same computer:

1. Log on to the computer you have selected for administrative tasks and browse to the location where you have saved downloaded Delinea files.

   If you have a physical CD that you made from the ISO image file, the Getting Started page is displayed automatically. If the page is not displayed, open the autorun.exe file to start the installation of Delinea software.

2. On the Getting Started page, click **Audit & Monitor** to start the setup program for audit and monitoring service service components.

   In the rare case where the administrator should not have access to the Audit Analyzer, select Audit Manager, then click **Next**.

After you install Audit Manager, you are prompted to create a new installation. If you want to create the installation at a later time, you can run the setup program again to create a new installation.

Creating a New Installation

Before you can begin audit and monitoring service, you must create at least one installation and a management database. Creating the management database, however, requires SQL Server system administrator privileges on the computer that hosts the SQL Server instance. If possible, you should have a database administrator add your Active Directory domain account to the SQL Server system administrators role.

If you have not been added to the system administrators role, you should contact a database administrator to assist you. For more information about creating a new installation when you don't have system administrator privileges, see How to create an installation without system administrator privileges.

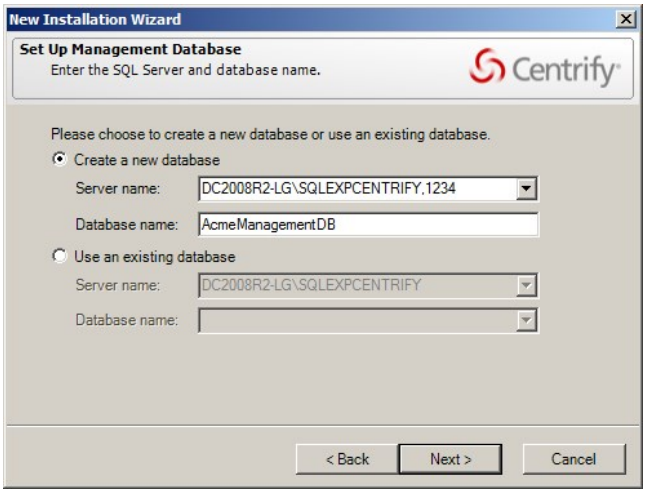
To create a new installation and management database as a system administrator:

1. Log on using an Active Directory account with permission to install software on the local computer.
2. Open the Audit Manager console to display the New Installation wizard.

   The New Installation wizard displays automatically the first time you start Audit Manager. You can also start it by clicking **Action > New Installation** or from the right-click menu when you select the Audit Manager node.
3. Type a name for the new installation, then click **Next**.

   Tip: {/b}Name the installation to reflect its administrative scope. For example, if you are using one installation for your entire organization, you might include the organization name and All or Global in the installation name, such as AcmeAll. If you plan to use separate installations for different regions or divisions, you might include that information in the name, for example AcmeBrazil for a regional installation or AcmeFinance for an installation that audits computers in the Finance department.
4. Select the option to create a new management database and verify the SQL Server computer name, instance name, and database name are correct, then click **Next**.

   If the server does not use the default TCP port (1433), you must provide the server and instance names separated by a backslash, then type a comma and the appropriate port number. For example, if the server name is ACME, the instance name is BOSTON, and the port number is 1234, the server name would be ACME\BOSTON,1234.



   If you're connecting to a SQL Server availability group listener, click Options (next to the Server Name) and enter the following connection string parameters:

   MultiSubnetFailover=Yes
5. Type the license key you received, then click **Add** or click **Import** to import the keys directly from a file, then click **Next**.
6. Accept the default location or click **Browse** to select a different Active Directory container to which you want to publish audit-related information, then click **Next**.
7. Select **Enable video capture recording of user activity** if you want to capture a full video record of desktop activity on Windows computers when users are audited, then click **Next**.

Selecting this option enables you to review everything displayed during an audited user session, but will increase the audit store database storage requirements for the installation. You can deselect this option if you are only interested in a summary of user activity in the form of audit trail events. Audit trail events are recorded when users log on, open applications, and select and use role assignments with elevated rights.

8. Review details about the installation and management database, then click **Next**.

   If you have SQL Server system administrator (sa) privileges and can connect to the SQL Server instance, the wizard automatically creates the management database.

9. Select the **Launch Add Audit Store Wizard** option if you want to start the Add Audit Store wizard, then click **Finish**.

   If you want to create the first audit store database at a later time, you should deselect the **Launch Add Audit Store Wizard** option and click **Finish**.

   For more information about adding the first audit store database, see Create the first audit store.

How to Create an Installation without System Administrator Privileges

If you do not have the appropriate permission to create SQL Server databases, you cannot use the New Installation wizard to create the management database without the assistance of a database administrator.

If you do not have system administrator privileges, the wizard prompts you to specify another set of credentials or generate SQL scripts to give to a database administrator. For example:

If you don't have a database administrator immediately available who can enter the credentials for you, you cannot continue with the installation.

To create an installation when you don't have system administrator privileges:

1. Select the option to generate the SQL scripts, then click **Next**.
2. Select the folder location for the scripts, then click **Next**.
3. Review details about the installation and management database you want created, then click **Next**.

   The wizard generates two scripts: Script1 prepares the SQL Server instance for the management database and Script2 creates the database.

4. Click **Finish** to exit the New Installation wizard.
5. Send the scripts to a database administrator with a service or change control request.

   Note: {/b}You should notify the database administrator that the scripts must be run in the proper sequence and not modified in any way. Changes to the scripts could render the database unusable.

6. After the database administrator creates the database using the scripts, open the Audit Manager console to run the New Installation wizard again.
7. Type the name of the installation, then click **Next**.

8. Select **Use an existing database** and verify the database server and instance name, then click the Database name list to browse for the database name that the database administrator created for you.

   If the server does not use the default TCP port, specify the port number as part of the server name. For example, if the port number is 1234, the server name would be similar to ACME\BOSTON,1234.

9. Select the database name from the list of available databases, click **OK**, then click **Next**.
10. You should only select an existing database if the database was created using scripts provided by Delinea.
11. Type a license key or import licenses from a file, then click **Next**.
12. Review details about the audit management database to be installed, then click **Next**.
13. Select the **Launch Add Audit Store Wizard** option if you want to start the Add Audit Store wizard, then click **Finish**.

Create the First Audit Store

If you selected the Launch Add Audit Store Wizard at the end of the New Installation Wizard, the Add Audit Store Wizard opens automatically. You can also open the wizard at any time by right-clicking the Audit Stores node in the Audit Manager console and choosing Add Audit Store.

To create the first audit store:

1. Type a display name for the audit store, then click **Next**.

   Tip: {/b}If your plan specifies multiple audit stores, use the name to reflect the sites or subnets serviced by this audit store. Note that an audit store is actually a record in the management database. It is not a separate process running on any computer. You use a separate wizard to create the databases for an audit store.

2. Click **Add Site** or **Add Subnet** to specify the sites or subnets in this audit store.
   - If you select Add Site, you are prompted to select an Active Directory site.
   - If you select Add Subnet, you are prompted to type the network address and subnet mask.

     After you make a selection or type the address, click **OK**. You can then add more sites or subnets to the audit store. When you are finished adding sites or subnets, click **Next** to continue.

     The computer you use to host the audit store database should be no more than one gateway or router away from the computers being audited. If your Active Directory sites are too broad, you can use standard network subnets to limit the scope of the audit store.

3. Review information about the audit store display name and sites or subnets, then click **Next**.
4. Select the **Launch Add Audit Store Database Wizard** option if you want to create the first audit store database, then click **Finish**.

Create the Audit Store Database

If you selected the Launch Add Audit Store Database Wizard check box at the end of the Launch Add Audit Store Wizard, the Add Audit Store Database Wizard opens automatically. You can also open the wizard at any time from the Audit Manager console by expanding an audit store, right-clicking the Databases node, and choosing Add Audit Store Database.

To create the first audit store database:

1. Type a display name for the audit store database, then click **Next**.

   The default name is based on the name of the audit store and the date the database is created.

2. Select the option to create a new database and verify that the SQL Server computer name, instance name, and database name are correct.

   The default database name is the same as the display name. You can change the database name to be different from the display name, if you want to use another name.

   If the server does not use the default TCP port, specify the port number as part of the server name. For example, if the port number is 1234, the server name would be similar to ACME\BOSTON,1234.

When entering the SQL Server host computer name, note that you can enter either the server short name (which is automatically resolved to its fully qualified domain name, or FQDN) or the actual server FQDN or the CNAME alias for the server.

If the database is an Amazon RDS SQL Server:

1. Select the **This is an Amazon RDS SQL Server** option.
2. In the Server Name field, enter the RDS SQL Server database instance endpoint name used for Kerberos authentication.

   For example, if the database host name is northwest1 and the domain name is sales.acme.com, then the endpoint name would be northwest1.sales.acme.com.

   Click **Options** to enter additional connection string parameters or to enable data integrity checking.

- You can enable or disable data integrity checking once, when you create the audit store database. To change the state, you must rotate to a new audit store database.

Connecting to SQL Server on a Remote Computer

To create an audit store database on a remote computer, there must be a one-way or two-way trust between the domain of the computer on which you are running the Add Audit Database wizard and the domain of the computer hosting SQL Server. The Active Directory user account that you used to log on to the computer where the Audit Manager is installed must be in a domain trusted by the computer running SQL Server. If there is no trust relationship, you must log on using an account in the same domain as the computer running SQL Server. If you are accessing the computer running SQL Server remotely, you can use the Run As command to change your credentials on the computer from which you are running the wizard.

Verify Network Connectivity

The computer hosting the SQL Server database for the active audit store server be online and accessible from the Audit Manager console and from the clients in the Active Directory site or the subnet segments you have defined for the audit store. You should verify that there are no network connectivity issues between the computers that will host collectors and those hosting the SQL Server databases.

How to Create the Database without System Administrator Privileges

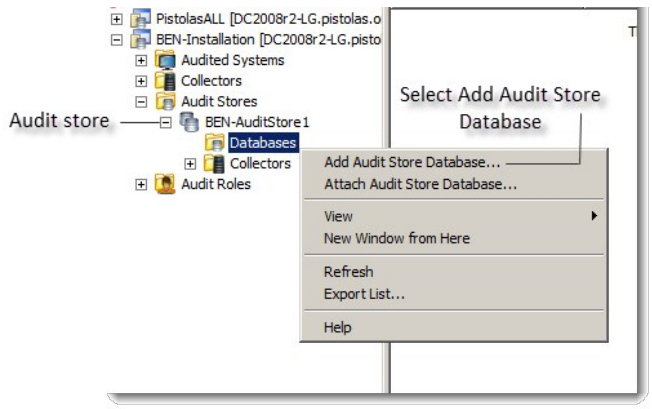
If you do not have system administrator privileges, the wizard prompts you to specify another set of credentials or generate SQL scripts to give to a database administrator. If you don't have database administrator credentials or a database administrator immediately available who can enter the credentials for you, you should generate the scripts, then follow the prompts displayed to exit the wizard.

To add the database to the audit store after you have generated the scripts:

1. Send the scripts to a database administrator with a service or change control request.

   Note: {/b}You should notify the database administrator that the scripts must be run in the proper sequence and not modified in any way. Changes to the scripts could render the database unusable.

2. After the database administrator creates the database using the scripts, open the Audit Manager console.
3. Expand the installation node, then expand Audit Stores and the specific audit store you for which you want a new database.
4. Select **Databases**, right-click, then click **Add Audit Store Database**. For example:



5. Type a display name for the audit store database, then click **Next**.
6. Select **Use an existing database** and select the database that the database administrator created for you.

   Because this is the first audit store database, you also want to make it the active database. This option is selected by default. If you are creating the database for future use and don't want to use it immediately, you can deselect the **Set as active database** option.

   If the server does not use the default TCP port, specify the port number as part of the server name. For example, if the port number is 1234, the server name would be similar to ACME\BOSTON,1234.

The installation, management database, and first audit store database are now ready to start receiving user session activity. Next, you should install the collectors and, finally, the agents to complete the deployment of the audit and monitoring service infrastructure.

Installing and Configuring Audit Collectors

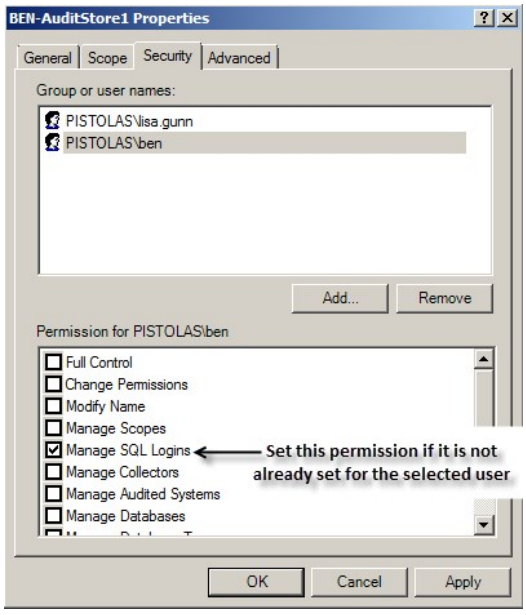
After you have created a new installation, with an audit management database and at least one audit store and audit store database, you must add the collectors that will receive audit records from the agents and forward those records to the audit store. For redundancy and scalability, you should have at least two collectors. For more information about planning how many collectors to use and the recommended hardware and network configuration for the collector computers, see Decide where to install collectors and audit stores.

Set the Required Permission

Before you configure a collector, you should check whether your user account has sufficient permissions to add new collector accounts to the audit store database. If you are a database administrator or logged on with an account that has system administrator privileges, you should be able to configure the collector without modifying your account permissions. If you have administrative rights on the computer hosting Audit Manager but are not a database administrator, you can set the appropriate permission before continuing.

To set the permission required to add accounts to the audit store database:

1. Open Audit Manager.
2. Expand the installation, then expand Audit Stores.
3. Select the audit store that the collector will connect to, right-click, then click **Properties**.
4. Click the **Security** tab.
5. Click **Add** to search for and select the user who will configure the collector.
6. Select the **Manage SQL Logins** right, then click **OK**.



Install the Collector Service Using the Setup Program

If your user account has sufficient permissions to add new collector accounts to the audit store database, you can install a collector by running the setup program on a selected computer. When prompted to select components, select Audit Collector and deselect all of the other components, then click **Next**. Follow the instructions in the wizard to select the location for installing files and to confirm your selections, then click **Finish** to complete the installation.

Configure the Audit Collector Service

By default, when you click **Finish**, the setup program opens the Collector Configuration Wizard. Alternatively, you can start the configuration wizard at any time by clicking Configure in the Collector Control Panel.

To configure the collector service:

1. Type the port number to use, then click **Next**.

   The default port is 5063 for communication from agents to the collector. If you want to use a different port, the wizard checks whether the port is open in the Windows firewall.

   If you're running another firewall product, open the port with the tools provided by that product. If there's an upstream firewall—such as a dedicated firewall appliance—between the Collector and the computers to be audited, contact the appropriate personnel to open the port on that firewall.

2. Select the installation of which this collector will be a part, then click **Next**.

   The configuration wizard verifies that the installation has an audit store that services the site that the collector is in and that the collector and its audit store database are compatible.

3. Select whether you want to use Windows authentication or SQL Server authentication when the collector authenticates to the audit store database, then click **Next**.

   In most cases, you should choose Windows authentication to add the computer account to the audit store database as a trusted, incoming user.

   If Microsoft SQL server is in a different forest or in an untrusted forest, you should use SQL Server Management Studio to set up one or more SQL Server login accounts for the collector. After you create the SQL Server login account for the collector to use, you can select SQL Server authentication, then type the SQL Server login name and password in the wizard.

4. Choose the maximum number of connections you want for the SQL Server Connection Pool, then click **Next**.

5. Review your settings for the collector, then click **Next**.

6. Click **Finish** to start the collector service and close the wizard.

Installing the Agent for Windows

You must install an agent on every Windows computer that you want to manage or audit. You can install the agent in the following ways:

- Interactively, by running the setup program on each computer.

  When the installation finishes, the agent configuration panel launches automatically. You can configure the agent to enable Delinea services right away, or exit the configuration panel and configure the agent later. See Installing the Agent for Windows interactively using the setup program for details about this installation method.

- Silently, by executing appropriate commands in a terminal window on each computer. This method also requires you to configure the agent registry settings on each computer. See Installing the Agent for Windows silently on remote Windows computers for details about this installation method.

  A variation of this method is to use a third-party software distribution product, such as Microsoft System Center Configuration Manager (SCCM), to execute the appropriate command line remotely, so that the software is deployed on remote computers. Using a third-party software distribution product is not covered in this guide.

- Silently and centrally, by using a Windows group policy to execute installation and registry configuration commands remotely on each computer that is joined to the domain. See Installing the Agent for Windows silently on all domain computers by using group policy for details about this installation method.

Regardless of the deployment method you choose, you should first make sure that the computers where you plan to deploy meet all of the installation prerequisites.

Verifying Prerequisites

Before installing the Agent for Windows, verify the computer on which you plan to install meets the following requirements:

- The computer is running a supported Windows operating system version.
- The computer is joined to Active Directory.
- The computer has sufficient processing power, memory, and disk space for the agent to use.
- The computer has Windows security update KB3033929 installed if it is running Windows 7 with Service Pack 1 or Windows Server 2008 R2 with Service Pack 1.
- The computer has .NET 4.6.2 or later installed.
- The computer has Windows Installer version 4.5 or newer.

If you are installing interactively using the setup program, the setup program can check that the local computer meets these requirements and install any missing software required. if you are installing silently or from a Group Policy Object, you should verify the computers where you plan to install meet these requirements.

Installing the Agent Interactively Using the Setup Program

The procedure in this section describes how to use the agent installation wizard to install the agent on a Windows computer. After the agent is installed, you will enable the agent to use one or more services that you installed earlier on the main administrative computer as described in Installing Server Suite and updating Active Directory.

To install the agent on Windows using the setup program:

1. Insert the distribution CD into the computer on which you wish to install the agent or browse to the location where you have saved downloaded files.
2. On the Getting Started page, click **Agent** to start the setup program for the agent.

   If the Getting Started page is not displayed, open the autorun.exe file to start the installation of Delinea software.
3. If a previous version of the agent is installed, click **Yes** when prompted to upgrade the Agent for Windows.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the License Agreement**, then click **Next**.
6. Accept the default location for installing components, or click **Change** to select a different location, then click **Next**.

7. In the Ready to install Agent for Windows page, click **Install**.
8. Click **Finish** to complete the installation and start the agent configuration panel.

   Go to Configuring the agent for details about using the agent configuration panel to enable Delinea services and configure how the agent interacts with those services.

Configuring the Agent

By default, when you click **Finish**, the setup program opens the agent configuration panel. In the agent configuration panel, you can enable the agent to connect to Delinea services that are installed on the main administrative computer as described in Installing Server Suite and updating Active Directory. After a service is enabled, you can use the agent configuration panel to configure settings that define how the agent will interact with each service.

The first time the agent configuration panel opens, it does not display any services for you to enable. Services display in the agent configuration panel only after you manually instruct the configuration panel to check for services and display those that are eligible to be enabled.

Only services that are installed and configured as required are eligible to be enabled. For example, if you installed the Privilege Elevation Service earlier (as described in Running the setup program on a Windows computer) but did not create a zone, the Privilege Elevation Service does not display on the list of services that you can enable.

To enable services using the agent configuration panel:

1. If the agent configuration panel is not open, open it by clicking **Agent Configuration** in the list of applications in the Windows Start menu.
2. In the agent configuration control panel, click **Add service**.

   All Delinea services that are available to be enabled are displayed.
3. In the list of Delinea services, highlight a service and click **OK**.
4. Provide additional information about the service that you are enabling:
   - **Audit & Monitoring Service**:

     In the Select an Audit Installation page, select an audit store from the list of available audit stores. Click **Next**, and the computer is connected to the audit store.
   - **Identity Platform Settings**:
     1. In the Connect to Identity Platform page, type the URL of the identity platform instance to connect to, or select an instance from the list of registered platform instances in the forest. Click **Next**.
     2. In the Multi-factor authentication for Windows Login page, ensure that the check box to enable multi-factor authentication is selected. Next, use the **All Active Directory accounts** button or **Accounts below** button to specify which Active Directory accounts are enabled for multi-factor authentication login. If you select **Account below**, use the **Add** and **Remove** buttons to select accounts. Click **Next** when you are finished.
   - **Privilege Elevation Service**:
     1. In the Join to a zone page, type a zone or select a zone from the list of available zones. You can also choose to select the option to retrieve the zone data before the computer restarts. This option can be helpful in situations where you might lose connection to the domain after restarting, such as when you're using a VPN connection.

        Click **Next**, and the computer is joined to the zone.
     2. After the computer is joined to a zone, you must reboot the computer to activate all privilege elevation service features on the computer.

        If the zone that you select is already configured with a Privileged Access Service tenant, the message **Identity Platform enabled** displays after the computer joins the zone. In this situation, the instance is managed by the zone, and is shown as read-only.
5. To add additional services, click **Add service** and repeat the preceding steps.

   When you are done, the services that you enabled are shown in the **Enabled services** section of the agent configuration panel.
6. If necessary, continue to configure Delinea services after their initial configuration during enablement as described in these sections:

- Configuring agent settings for the audit and monitoring service
- Configuring agent settings for offline audit and monitoring service storage
- Configuring agent settings for the Identity Platform
- Configuring agent settings for privilege elevation

Configuring Agent Settings for the Audit and Monitoring Service

If you want to reconfigure agent settings for auditing on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the service), you can open the agent configuration panel manually and configure the agent as described in this section.

To configure agent settings for audit and monitoring service:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

   The agent configuration panel opens, and displays the Delinea services that are currently enabled. You can configure any service listed in the **Enabled services** section.

2. Click **Audit & Monitoring Service**, and then click **Settings**.
3. In the General tab, click **Configure**.
4. Select the maximum color quality for recorded sessions, then click **Next**.

   See Selecting the maximum color quality for recorded sessions for more information on the configuration of this setting.

5. Specify the offline data location and the maximum percentage of disk that the offline data file should be allowed to occupy, then click **Next**.

   See Configuring agent settings for offline audit and monitoring service storage for more information on the configuration of this setting.

6. Select the installation that the agent belongs to, then click **Next**.
7. Review your settings, then click **Next**.
8. Click **Finish**.
9. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see Monitoring collector status locally.

Selecting the Maximum Color Quality for Recorded Sessions

Because auditing Windows computers captures user activity as video, you can configure the color depth of the sessions to control the size of data that must be transferred over the network and stored in the database. A higher color depth increases the CPU overhead on audited computers but improves resolution when the session is played back. A lower color depth decreases network traffic and database storage requirements, but reduces the resolution of recorded sessions.

The default color quality is low (8-bit).

Configuring Agent Settings for Offline Audit and Monitoring Service Storage

The “Maximum size of the offline data file” setting defines the minimum percentage of disk space that should be available, if needed, for audit and monitoring service. It is intended to prevent audited computers from running out of disk space if the agent is sending data to its offline data storage location because no collectors are available.

For example, if you set the threshold to 10%, auditing will continue while spooling data to the offline file location as long as there is a least 10% of available disk space on the spool partition. When the available disk space reaches the threshold, auditing will stop until a collector is available.

The agent checks the spool disk space by periodically running a background process. By default, the background process runs every 15 seconds. Because of the delay between background checks, it is possible for the actual disk space available to fall below the threshold setting. If this were to occur, auditing would stop at the next interval. You can configure the interval for the background process to run by editing the

HKLM\Software\Centrify\DirectAudit\Agent\DiskCheckInterval registry setting.

Configuring Agent Settings for the Identity Platform

If you want to reconfigure agent settings for the Identity Platform on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the service), you can open the agent configuration panel manually and configure the agent as described in this section.

To configure agent settings for the Identity Platform:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

   The agent configuration panel opens, and displays the Delinea services that are currently enabled. You can configure any service listed in the **Enabled services** section.

2. Click **Identity Platform**, and then click **Settings**.

3. In the General tab, review the authentication options in the Features area:

   - **Multi-factor authentication:**

     - If the status is **Enabled**, the computer is not joined to a zone, and you can configure all Identity Platform settings that are shown in the General tab.

     - If the status is **Enabled per zone settings**, the computer is joined to a zone, and most Identity Platform settings are based on the zone configuration.

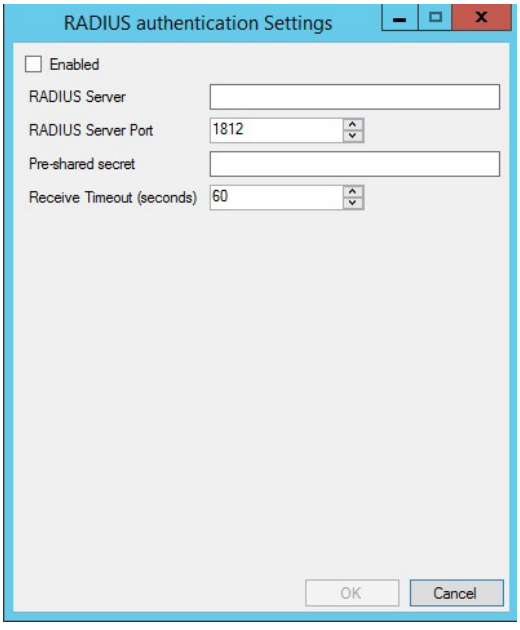
       In this situation, the **Browse** and **Details** buttons in the General tab are disabled, because those features are controlled by the zone configuration. The only configuration that you can perform in the General tab is to change the proxy server settings.

       Multi-factor authentication displays in the Authentication Source drop-down once the status is Enabled per zone settings.

   - **RADIUS authentication:**

     - If the status is **Enabled**, you can select this option to use as the authentication option for privilege elevation. You can enable this option either by group policy or a local configuration setting.

     - If the status is **Disabled**, click **Details** to configure and enable the RADIUS server connection.

       RADIUS authentication displays in the Authentication Source drop-down once the status is Enabled.

4. To change proxy server settings:
    1. Click **Change**.
    2. Specify a new proxy server address.
    3. Click **OK**.
5. To change to a different Identity Platform instance (only configurable if the computer is not joined to a zone):
    1. Click **Browse**.
    2. Select an instance from the list of registered platform instances in the forest.
    3. Click **OK**.
6. To specify which Active Directory accounts require multi-factor authentication (only configurable if the computer is not joined to a zone):
    1. Click Details.
    2. Use the **All Active Directory accounts** button or **Accounts below** button to specify which Active Directory accounts are enabled for multi-factor authentication login. If you select **Account below**, use the **Add** and **Remove** buttons to select accounts.
    3. Click **OK**.
7. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see the *Multi-factor Authentication Quick Start Guide*.

Configuring Agent Settings for Privilege Elevation

If you want to reconfigure agent settings for privilege elevation on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the service), you can open the agent configuration panel manually and configure the agent as described in this section.

If you haven't yet configured the agent settings for privilege elevation, see Configuring the agent for details.

To configure existing agent settings for privilege elevation:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

    The agent configuration control panel opens, and displays the services that are currently enabled. You can configure any service listed in the **Enabled services** section.
2. Click **Privilege Elevation Service**, and then click **Settings**.
3. In the General tab, click **Change**.
4. In **Change the zone for this computer**, click **Browse**.
5. Click **Find Now** to search for an appropriate zone for the agent.
6. Select a zone from the list of search results, then click **OK**.
7. Click **OK** to use the zone you selected.
8. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see Running diagnostics and viewing logs for the agent.

Installing the Agent without MFA Login

If desired, you can install the Agent for Windows without the MFA login feature. This can be useful in situations where either you don't want to enforce multi-factor authentication or you don't use Privileged Access Service.

To install the Agent for Windows without the MFA login feature:

- Run the following command:

msiexec /i "Agent for Windows64.msi" /qn PRIVILEGEONLY=1

Installing the Agent for Windows Silently on Remote Windows Computers

If you want to perform a "silent" (also called *unattended*) installation of the Agent for Windows, you can do so by specifying the appropriate command line options and Microsoft Windows Installer (MSI) file to deploy. You must execute the commands on every Windows computer that you want to manage or audit.

Note: {/b}You can also use a silent installation to automate the installation or upgrade of the agent on remote computers if you use a software distribution product, such as Microsoft System Center Configuration Manager (SCCM), to deploy software packages. However, installing remotely in this way is not covered in this topic.

Deciding to Install with or without Joining the Computer to a Zone

Before you begin a silent installation, you should decide whether you will wait until later to join the computer to a zone, or join the computer to a zone as part of the installation procedure.

If you install without joining a zone during installation:

- See Configuring registry settings for details about the registry settings that you can configure manually after the installation finishes.
- See Installing silently without joining a zone for details about performing the installation.

If you install and join a zone during installation:

- You use a transform (MST) file that is provided with Server Suite to configure a default set of agent-specific registry keys during the silent installation.
- You can optionally edit the MST file before performing the installation to customize agent-specific registry settings for your environment.
- You can optionally use the registry editor to configure registry settings after the installation finishes.
- See Configuring registry settings for details about the registry settings that you can configure by editing the MST file.
- See Editing the default transform (MST) file for details about how to edit the MST file before you perform the installation.
- See Installing and joining a zone silently for details about performing the installation.

Configuring Registry Settings

When you perform a silent installation, several registry settings specific to the agent are configured by the default MSI file. In addition, a default transform (MST) file is provided for you to use if you join the computer to a zone as part of the installation procedure. When executed together, the default MSI and MST files ensure that the computer is joined to a zone, and that a default set of agent-specific registry keys is configured.

If your environment requires different or additional registry settings, you can edit the MST file before performing an installation. Then, when you execute the MSI and MST files to perform an installation, your customized registry settings are implemented. For details about how to edit the MST file, see Editing the default transform (MST) file.

> **Note**: If you do not join the computer to a zone during installation, you do not use the MST file. In this situation, you can create or edit registry keys manually after the installation finishes by using the registry editor.

The following table describes the agent-specific registry settings that are available for you to configure during installation (by using the MST file) or after installation (by using the or the registry editor). Use the information in this table if you need to configure registry settings differently than how they are configured by the default MSI and MST files. Keep the following in mind as you review the information in the table:

- The default MSI file is named Agent for Windows64.msi, and is located in the **Agent** folder in the Delinea download location.
- The default MST file is named Group Policy Deployment.mst, and is located in the **Agent** folder in the Delinea download location.
- If you want to install the agent without the MFA login feature, use the Group Policy Deployment-PrivilegeOnly.mst, and is located in the **Agent** folder in the Delinea download location.
- All of the settings in the following table are optional, although some are included in the default MSI and MST files so that they are configured when the MSI and MST files execute during an installation.
- Settings that are included in the default MSI and MST files are noted in the table.
- Some settings are environment-specific, and therefore do not have a default value. Others are not environment-specific, and do have a default value.
- The settings described in the table are located in the MSI file's Property table.
- The **Setting** column shows both the property name in the MSI file, and the name (in parentheses) of the registry key in the Windows registry.

	Specifies the color depth of sessions recorded by the agent. The color depth affects the resolution of the activity recorded and the size of the records stored in the audit store database when you have video

Auditing and Monitoring	REG_MAX_FORMAT (MaxFormat)	capture auditing enabled. You can set the color depth to one of the following values: 0 to use the native color depth on an audited computer. 1 for a low resolution with an 8-bit color depth 2 for medium resolution with a 16-bit color depth (default) 4 for highest resolution with a 32-bit color **This setting is included in the default MSI file.** In the registry, this setting is specified by a numeral (for example, 1). In the MSI file Property table, it is specified by the # character and a numeral (such as #1). The default value is 1.
Auditing and Monitoring	REG_DISK_CHECK_THRESHOLD (DiskCheckThreshold)	Specifies the minimum amount of disk space that must be available on the disk volume that contains the offline data storage file. You can change the percentage required to be available by modifying this registry key value. **This setting is included in the default MSI file.** In the registry, this setting is specified by a numeral (for example, 1). In the MSI file Property table, it is specified by the # character and a numeral (such as #10). The default value is 10, meaning that at least 10% of the disk space on the volume that contains the offline data storage file must be available. If this threshold is reached and there are no collectors available, the agent stops spooling data and audit data is lost.
Auditing and Monitoring	REG_SPOOL_DIR (SpoolDir)	Specifies the offline data storage location. The folder location you specify will be where the agent saves ("spools") data when it cannot connect to a collector. **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that it is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Auditing and Monitoring	REG_INSTALLATION_ID (InstallationId)	Specifies the unique global identifier (GUID) associated with the installation service connection point. **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that it is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Auditing and Monitoring	REG_LOG_LEVEL_DA (LogLevel)	Specifies what level of information, if any, is logged. Possible values are: off information warning error verbose **This setting is included in the default MSI file.** The default value is information.
Authentication & Privilege	REG_RESCUEUSERSIDS (RescueUserSids)	Specifies which users have rescue rights. Type user SID strings in a comma separated list. For example: *user1SID,user2SID,usernSID* **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that the setting is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Authentication & Privilege	REG_LOG_LEVEL_DZ (LoggingLevel)	Specifies what level of information, if any, is logged. Possible values are: off information warning error verbose **This setting is included in the default MSI file.** The default value is information.
Authentication & Privilege	GPDeployment	Specifies whether the computer is joined to the zone where the computer was pre-created. This setting is used only during installation and does not have a corresponding registry key. Possible values are: 0 - The computer is not joined to the zone. 1 - The computer is joined to the zone. **This setting is included in the default transform (MST) file.** To use it, you must execute the MST file when you execute the default MSI file. The default value is 1, meaning that the pre-created computer is joined to the zone.
Authentication & Privilege	ZONEDATA	Specifies the option to retrieve the zone data before the computer restarts. This option can be helpful in situations where you might lose connection to the domain after restarting, such as when you're using a VPN connection. Possible values are: YES NO The default value is NO in the default MSI file.

Editing the Default Transform (MST) File

This section describes how to edit the default transform (MST) file Group Policy Deployment.mst. You execute the MST file together with the installation (MSI) file during a silent installation if you want to join the computer to a zone as part of the installation.

The MST file specifies registry key settings that are different from those specified in the MSI file. You use the MST file to customize a silent installation for a specific environment. Using an MST file makes it unnecessary to edit registry keys manually after a silent installation.

Note: {/b}By default, auditing features are installed when you install the Agent for Windows. The service is not enabled by default, but the service item in the configuration panel appears if the feature is enabled through group policy.

See Installing and joining a zone silently for instructions about how and when to execute the MST file.

To edit the default MST file:

1. You will use the Orca MSI editor to edit the MST file. Orca is one of the tools available in the Windows SDK. If the Windows SDK (or Orca) is not installed on your computer, download and install it now from this location:

   https://msdn.microsoft.com/en-us/library/aa370557(v=vs.85).aspx

2. Execute Orca.exe to launch Orca.
3. In the **Agent** folder in the Delinea download location, copy Group Policy Deployment.mst so that you have a backup.
4. In Orca, select **File > Open** and open the Agent for Windows64.msi file located in the **Agent** folder in the Delinea download location.
5. In Orca, select **Transform > Apply Transform**.
6. In Orca, navigate to the **Agent** folder in the Delinea download location and open Group Policy Deployment.mst.

   The file is now in transform edit mode, and you can modify data rows in it.

7. In the Orca left pane, select the Property table.

   Notice that a green bar displays to the left of "Property" in the left pane. This indicates that the Property table will be modified by the MST file.

   The right pane displays the properties that configure registry keys when the MSI file executes. Notice that the last property in the table, GPDeployment, is highlighted in a green box. This indicates that the GPDeployment property will be added to the MSI file by the MST file.

   Note: {/b}In order for the computer to join a zone during installation, the Group Policy Deployment.mst file *must* specify the GPDeployment property with a value of 1.

8. In the right pane, edit or add properties as necessary to configure registry keys for your environment. See the table in Configuring registry settings for details about agent-specific properties that are typically set.

   - To edit an existing property, double click its value in the **Value** column and type a new value.
   - To add a new property, right-click anywhere in the property table and select **Add Row**.

9. After you have made all necessary modifications, select **Transform > Generate Transform** to save your modifications to the default MST file.

   Be sure to save the MST file in the same folder as the MSI file. If the MST and MSI files are in different folders, the MST file will not execute when you execute the MSI file.

   The MST file is now ready to be used as described in Installing and joining a zone silently.

Installing Silently without Joining a Zone

This section describes how to install the agent silently without joining the computer to a zone. This procedure includes configuring registry settings manually using the registry editor or a third-party tool.

Note: {/b}To install the agent and join the computer to a zone during installation, see Installing and joining a zone silently for more information.

Check prerequisites:

1. Verify that the computers where you plan to install meet the prerequisites described in Verifying prerequisites. If prerequisites are not met, the silent installation will fail.
2. If you are installing audit and monitoring service, verify that the following tasks have been completed:
   1. Installed and configured the SQL Server management database and the SQL Server audit store database.
   2. Installed and configured one or more collectors.
   3. Configured and applied the DirectAudit Settings group policy that specifies the installation name.

To install the Agent for Windows silently without joining the computer to a zone:

1. Open a Command Prompt window or prepare a software distribution package for deployment on remote computers.

   For information about preparing to deploy software on remote computers, see the documentation for the specific software distribution product you are using. For example, if you are using Microsoft System Center Configuration Manager (SCCM), see the Configuration Manager documentation.

2. Run the installer for the Agent for Windows package. For example:

   msiexec /qn /i "Agent for Windows64.msi"

   By default, none of the services are enabled.

3. Use the registry editor or a configuration management product to configure the registry settings for each agent. See the table in Configuring registry settings for details about agent-specific registry keys that you can set.

   For example, under

   HKEY_LOCAL_MACHINE\Software\Centrify\DirectAudit\Agent, you could set the DiskCheckThreshold key to a value other than the default value of 10%.

Installing and Joining a Zone Silently

This section describes how to install the agent and join the computer to a zone at the same time. The procedure described here includes the following steps in addition to executing the MSI file:

- You first prepare (pre-create) the Windows computer account in the appropriate zone.
- You execute an MST file together with the MSI file to join the computer to a zone and configure registry settings during the installation.

  Note: {/b}Joining the computer to a domain is applicable only when you are enabling Authentication & Privilege features.
  To install the agent without joining the computer to a zone during installation, see Installing silently without joining a zone for more information.

Check prerequisites:

1. Verify that the computers where you plan to install meet the prerequisites described in Verifying prerequisites. If prerequisites are not met, the silent installation will fail.
2. If you are enabling audit and monitoring service in addition to Authentication & Privilege, verify that the following tasks have been completed:
   1. Installed and configured the SQL Server management database and the SQL Server audit store database.
   2. Installed and configured one or more collectors.
   3. Configured and applied the DirectAudit Settings group policy that specifies the installation name.

To install the Agent for Windows and add a computer to a zone during installation:

1. Prepare a computer account in the appropriate zone using Access Manager or the PowerShell command New-CdmManagedComputer. See Preparing Windows computer accounts for more information.

2. You will use the default transform file Group Policy Deployment.mst in Step 3 to update the MSI installation file so that the computer is joined to the zone in which it was pre-created in Step 1. You can optionally modify Group Policy Deployment.mst to change or add additional registry settings during installation.

   If you want to edit Group Policy Deployment.mst to change or add additional registry settings and have not yet done so, edit it now as described in Editing the default transform (MST) file.

   In order for the computer to join the zone from Step 1, the Group Policy Deployment.mst file *must* specify the GPDeployment property with a value of 1.

3. Run the following command:

   msiexec /i "Agent for Windows64.msi" /qn TRANSFORMS="Group Policy Deployment.mst"

   By default, Privilege Elevation Service is enabled by joining a zone. If the zone is also configured with a platform instance (tenant), Identity Services Platform will also be enabled. If you want to enable auditing, configure the corresponding registry value in the Property page of the MST file: REG_CURRENT_INSTALLATION or via Group Policy.

   You can also choose to install the specify the option to retrieve the zone data before the computer restarts. This option can be helpful in situations where you might lose connection to the domain after restarting, such as when you're using a VPN connection. To specify that the agent retrieves zone

data before the computer restarts, run the following command:

msiexec /i "Agent for Windows64.msi" /qn TRANSFORMS="Group Policy Deployment.mst" ZONEDATA="YES"

The computer will be restarted automatically to complete the deployment and start the agent.

Installing the Agent for Windows Silently on All Domain Computers by Using Group Policy

You can use a group policy object (GPO) to automate the deployment of the Agent for Windows. Because automated installation fails if all the prerequisites are not met, be sure that all the computers on which you intend to install meet the requirements described in Verifying prerequisites.

> **Note**: If you install the Common Component before you install the agent, information about the installation of the agent can be captured in a log file for troubleshooting purposes.

To create a new group policy object for the deployment of the Agent for Windows:

1. Prepare computer accounts in the appropriate zones using Access Manager or the PowerShell command New-CdmManagedComputer. See Preparing Windows computer accounts for more information.
2. Copy the Agent for Windows64.msi and Group Policy Deployment.mst installer files to a shared folder on the domain controller or another location accessible from the domain controller.

   When you select a folder for the agent installer files, right-click and select **Share with > Specific people** to verify that the folder is shared with Everyone or with appropriate users and groups.
3. Right-click on the Agent for Windows64.msi file, then select **Edit with Orca**.
4. Select **Transform > Apply Transform**, then select Group Policy Deployment.mst from the same location as the Agent for Windows64.msi file.
5. Select the Property table on the left hand side and add the following:

REG_ZONELESS_MFA_TENANT	Tenant URL	(Ex: https://aaa1111.my.delinea.net:443/) Note: You must include “https://” and “:443/”.
REG_ZONELESS_MFA_ENABLED	true	Default Value = false
REG_EFFECTIVE_ZONELESS_MFA_USERS	Comma-Separated user or group names, or enter * for All AD users	
REG_CONNECTOR_BRANDING	Delinea	

1. Close Orca and save the changes as a new mst file.

   Make sure you save it in the same location as the msi file.
2. On the domain controller, click **Start > Administrative Tools > Group Policy Management**.
3. Select the domain or organizational unit that has the Windows computers where you want to deploy the Agent, right-click, then select **Create a GPO in this domain, and Link it here**.
4. For example, you might have an organizational unit specifically for Delinea-managed Windows computers. You can create a group policy object and link it to that specific organizational unit.
5. Type a name for the new group policy object, for example, Agent Deployment, and click **OK**.
6. Right-click the new group policy object and click **Edit**.
7. Expand **Computer Configuration > Policies > Software Settings**.
8. Select **Software installation**, right-click, and select **New > Package**.

9. Navigate to the folder you selected previously, then select the Agent for Windows64.msi file, and click **Open**.
10. Select **Advanced** and click **OK**.
11. Click the **Modifications** tab and click **Add**.
12. Select the .mst file created previously, then click **Open**, and click **OK**.
13. Close the Group Policy Management Editor, right-click the Agent Deployment group policy object, and verify that **Link Enabled** is selected.

    By default, when computers in the selected domain or organizational unit receive the next group policy update or are restarted, the agent will be deployed and the computer will be automatically rebooted to complete the deployment of the agent.

If you want to test deployment, you can open a Command Prompt window to log on to a Windows client as a domain administrator and force group policies to be updated immediately by running the following command:

gpupdate /force

After installation, all of the registry settings that were specified in the MSI and MST files are configured. If you need to further configure registry settings, use the registry editor to do so as described in Installing the Agent for Windows silently on remote Windows computers.

Installing the Agent on a Computer Running Server Core

You cannot use the autorun.exe or the setup.exe program to install components on a computer that is configured to run as a Server Core environment. Instead, you must install from Microsoft Installer (.msi) files using the msiexec command-line program.

To install the Agent for Windows on Server Core:

1. Use the Deployment Image Servicing and Management (DISM) or another command-line tool to enable the .NET Framework.

   For example, if the .NET Framework is located on the installation media in the D:\sources\sxs folder, use the following command:

   DISM /Online /Enable-Feature /FeatureName:NetFx3 /All /LimitAccess /Source:D:\sources\sxs

2. Copy the Agent for Windows files to the Server Core computer.

   For example:

   copy D:\Common\Centrify* C:\Centrify Agent

   copy D:\Agent* C:\Centrify Agent

3. Install the Common Component service using the .msi file.

   For example, to install the Common Component on a computer with 64-bit architecture, you might use the following command:

   msiexec /i Centrify Common Component64.msi" /qn

4. Install the Agent for Windows using the .msi file.

   For example, to install the Agent for Windows with identity management, privilege elevation, auditing, and monitoring features enabled on a computer with 64-bit architecture, you might run the following command:

   msiexec /qn /i "Agent for Windows64.msi" ADDLOCAL=ALL

   You can also choose to install the specify the option to retrieve the zone data before the computer restarts. This option can be helpful in situations where you might lose connection to the domain after restarting, such as when you're using a VPN connection. To specify that the agent retrieves zone data before the computer restarts, run the following command:

   msiexec /i "Agent for Windows64.msi" /qn TRANSFORMS="Group Policy Deployment.mst" ZONEDATA="YES"

5. Restart the computer with the appropriate shutdown options to complete the installation and start agent services.

   For example, you might run the following command:

   shutdown /r

Installing Additional Consoles

You can install additional consoles on any domain computers you want to use for managing access using zones or roles, or for managing the audit and monitoring service infrastructure. You also might want to install additional consoles on the computers to be used by auditors. You can install additional consoles from the Management Services setup program or from individual component-specific setup programs. For example, you can use the Audit Analyzer Console.exe setup program to install Audit Analyzer on a computer.

Installing Group Policy Extensions Separately from Access Manager

Group policy extensions are packaged separately from Access Manager, enabling the following installation options:

- You can install group policy extensions on any Windows domain computer without also installing Access Manager on the computer.
- You can install Access Manager on any Windows domain computer without also installing group policy extensions on the computer.

The group policy extension package has its own .exe and .msi installer files, so that you can install group policy extensions interactively through an installation wizard (by executing the .exe file) or silently from the command line (by executing the .msi file). Additionally, you can select or de-select the group policy extensions for installation when you run the Access Manager installation wizard.

Note: {/b}At the start of an installation, the group policy extension installer checks for previously installed versions of group policy extensions. If it detects a newer version than the version you are trying to install, the installation stops.

To install standalone group policy extensions interactively with the group policy installer:

1. On the Windows domain computer where you will install group policy extensions, navigate to the Delinea ISO bundle containing the group policy extension installer file.

   The installer file is named <!---TODO update filename---> CentrifyDC_GP_Extension-*#.#.#-architecture.*exe.

   For example: <!---TODO update filename--->

   CentrifyDC_GP_Extension-5.2.3-win64.exe

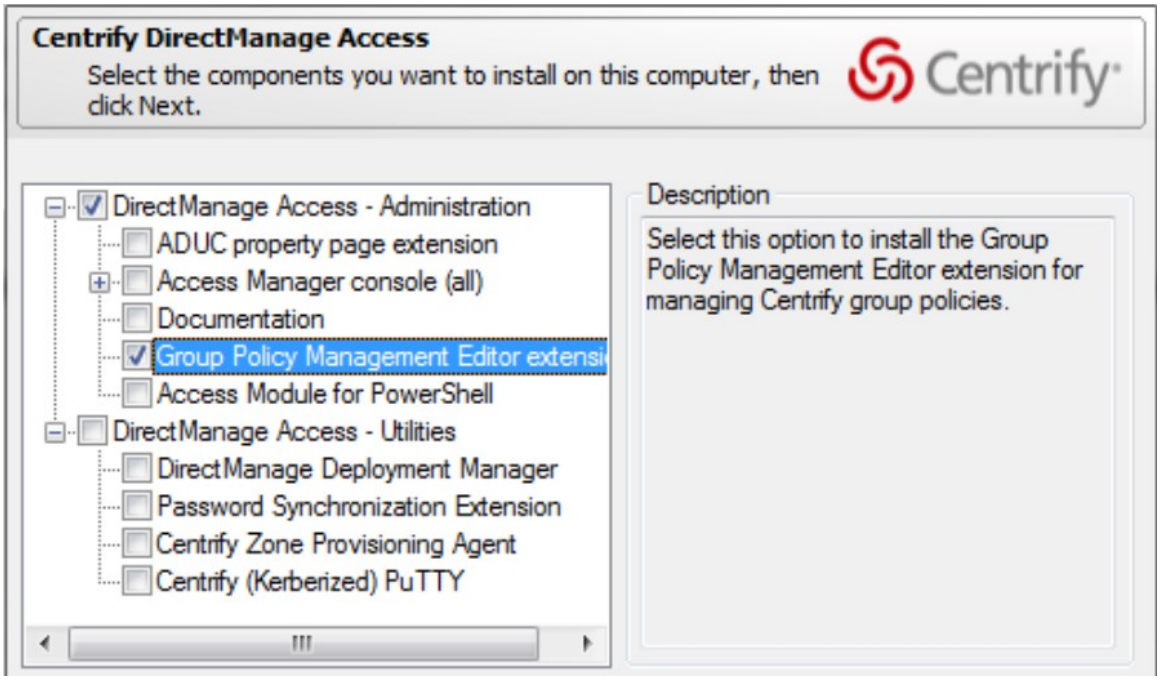
   In most distributions, the installer file is located in the following folder in the ISO bundle:

   DirectManage\Group Policy Management Editor Extension

2. Double-click the installer file to launch the Group Policy Management Editor Extension Setup Wizard.
3. Follow the wizard installation instructions to install the group policy extensions.

To install standalone group policy extensions interactively with the Management Services installer:

1. On the Windows domain computer where you will install group policy extensions, launch the setup program for Management Services components as described in Installing Server Suite and updating Active Directory.
2. Proceed through the setup program until you reach the wizard page in which to select individual components to install.
3. De-select every component except for **Group Policy Management Editor extension for managing group policies**:



4. Continue to follow the wizard installation instructions as described in Installing Server Suite and updating Active Directory until you are finished with the installation.

To install standalone group policy extensions silently without installing Access Manager:

1. Open a Command Prompt window.
2. Execute the group policy extension .msi installer file from the command line.

   The installer file is named <!---TODO update filename---> CentrifyDC_GP_Extension-*#.#.#-architecture.*msi.

   For example: <!---TODO update filename--->

   CentrifyDC_GP_Extension-5.2.3-win64.msi

   In most distributions, the installer file is located in the following folder in the ISO bundle:

   DirectManage\Group Policy Management Editor Extension

   The following is a typical command to run the 64-bit .msi installer file: <!---TODO update filename--->

   msiexec /qn /i "CentrifyDC_GP_Extension-5.2.3-win64.msi"

   For more information about installing with a .msi file, see Installing the Agent for Windows silently on remote Windows computers.

To install Access Manager interactively without installing group policies:

1. On the Windows domain computer where you will install group policy extensions, launch the Management Services setup program and select Authentication & Privilege as described in Installing Server Suite and updating Active Directory.
2. Proceed through the setup program until you reach the wizard page in which to select individual components to install.
3. De-select the **Group Policy Management Editor extension** component.
4. Continue to follow the wizard installation instructions as described in Installing Server Suite and updating Active Directory until you are finished with the installation.

Zones are the key component for organizing access rights and role assignments for Windows computers. This chapter describes how to use Access Manager to create zones, manage zone properties, add Windows computers to selected zones, and move and rename zone objects.

Starting Access Manager for the First Time

The first time you start Access Manager, a Setup Wizard prepares the Active Directory forest with parent containers for licenses and zones. The Setup Wizard also sets the appropriate permissions for the objects. For example, all authenticated users are granted read access of the Licenses container by default. These steps are typically performed once by a domain administrator. If you choose to, you can create the container objects manually.

What to do Before Updating Active Directory

Before you use Access Manager the first time, you should contact the Active Directory administrator to determine the appropriate location for the Licenses and Zones parent containers and whether you have the appropriate rights for completing this task. The specific administrative rights required for this task depend on the policies of your organization and who has permission to create classStore and parent and child container objects in Active Directory.

Rights Required for this Task

If you don't have administrative rights to create container objects in Active Directory, a domain administrator in the forest root domain can manually create the container objects and set the rights on those objects to allow other users to complete the initial configuration without being members of an administrative group.

The following table describes the minimum rights that must be granted on manually created container objects for other users to successfully complete the configuration with the Setup Wizard.

Licenses container	Read all properties Create classStore objects Modify permissions	This object only
	Write Description property Write displayName property	This object and all child objects
	By default, all Authenticated Users have read and list contents permission for the Licenses container and all of its child objects.	
Zones container	Read all properties Create classStore objects Create Container objects	This object only
	Write displayName property	This object and all child objects

If you are a domain administrator and use the Setup Wizard to create the container objects, you should add a security group for Zone Administrators to Active Directory. Set the following permissions on the parent Zones container to allow other users to manage zones.

Zones container	Read all properties Create Container objects Delete Container objects	This object only
	Write displayName property	This object and all child objects

Who Should Perform this Task

A Windows Active Directory administrator performs this task, depending on your organization's policies, by running the Setup Wizard or by manually creating container objects and notifying another user of the location of the container objects. The user who runs the Setup Wizard must be granted the rights required to create classStore objects.

How Often You Should Perform this Task

In most organizations, you only do this once for an Active Directory forest. However, if you want to create more than one administrative boundary, you can create additional parent containers as needed.

Steps for Completing this Task

The following instructions illustrate how to run the Setup Wizard from Access Manager.

To update Active Directory using Access Manager:

1. Open Access Manager.
2. At the Welcome page, click **Next**.
3. Select **Use currently connected user credentials** to use your current log on account or select **Specify alternate user credentials** and type a user name and password, then click **Next**.
4. Select a location for installing license keys in Active Directory, then click **Next**.

   The default container for license keys is <!---TODO update path ---> *domain_name*/Program Data/Centrify/Licenses. To create or select a container object in a different location, click **Browse**. If an Active Directory administrator has created the Licenses container for you, click Browse and navigate to the appropriate location. The Setup Wizard will create a classStore object in the location you specify.

   You can create additional containers in other locations later using the Manage Licenses dialog box.
5. Review the permission requirements for the container, then click **Yes** to confirm your selection.
6. Type or copy and paste the license key you received, then click **Add**.

   If you received multiple license keys, add each key to the list of installed licenses, then click **Next**. If you received license keys in a text file, click **Import** to import the keys directly from the file instead of adding the keys individually, then click **Next**.
7. Select **Create default zone container** and specify a location for the Zones container, then click **Next**.

   The default container location for zones is <!---TODO update ---> *domain_name*/Program Data/Centrify/Zones. To create or select a container object in a different location, click **Browse**. If an Active Directory administrator has created the Zones container for you, click Browse and navigate to the appropriate location. The Setup Wizard will create a classStore object in the location you specify.

   Any zones you create are placed in this container location by default.

   The next three pages only apply if you are managing multiple platforms. For a Windows-only deployment, you can click **Next** to leave the following options unselected:

   - Grant computer accounts in the Computers container permission to update their own account information.
   - Register administrative notification handler for Microsoft Active Directory Users and Computers snap-in.
   - Activate profile property pages.
8. Review and confirm your configuration settings, click **Next**, then click **Finish**.

After you click Finish, the Access Manager console is displayed.

What to Do Next

Create at least one parent zone.

Where you can find additional information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- Access control for Windows computers
- How zones organize access rights and roles
- Identity and privilege management

Preparing to Use Zones

One of the most important aspects of managing computers with Delinea software is the ability to organize computers, users, and groups into **zones**. You use zones to create logical groupings for:

- Managing access rights, role definitions, and role assignments.
- Delegating administrative tasks based on a separation of duties.
- Associating groups of computers and groups of users with specific role assignments.

Controlling Access through Hierarchical Zones

Server Suite for Windows only supports **hierarchical zones**. Hierarchical zones enable you to establish parent-child zone relationships, allowing rights, role definitions, and role assignments to be inherited down the zone hierarchy. One of the first decisions you need to make is how you can use the zone hierarchy most effectively.

With hierarchical zones, you define rights and roles in a parent zone so that those definitions are available in one or more child zones, as needed. Child zones can also inherit user and group role assignments. At any point in the zone hierarchy, you can choose to use or override information from a parent zone.

There are no predefined limits to the number of zones that can be used in a zone hierarchy or the number of levels deep zones can be nested in the hierarchy you define. For practical purposes, keep the hierarchy similar to the following:

- One or more top-level **parent** zones that includes all users and groups.
- One to three levels of intermediate **child** zones based on natural access control or administrative boundaries.

There are many different approaches you can take to defining the scope of a zone, including organizing by platform, department, manager, application, geographical location, or how a computer is used. The factors that are most likely to affect the zone design, however, will involve managing access rights and roles and delegating administrative tasks to the appropriate users and groups.

Managing Access Rights and Roles Using Zones

Zones enable you to grant specific rights to users in specific roles on specific computers. By assigning roles, you can control the scope of resources any particular group of users can access and what those users can do. For example, all of the computers in the finance department could be grouped into a single zone called “finance” and the members of that zone could be restricted to finance employees and senior managers, each with specific rights, such as permission to log on locally, access a database, update certain files, or generate reports.

Rights represent specific operations users are allowed to perform. A **role** is a collection of rights that can be defined in a parent or child zone and inherited. For example, a role defined in a parent zone can be used in a child zone, in a computer role, or at the computer level.

System and Predefined Rights

There are specialized login rights, called system rights. The system rights for Windows computers are:

- **Console login is allowed**: Specifies that users are allowed to log on locally using their Active Directory account credentials.
- **Remote login is allowed**: Specifies that users are allowed to log on remotely using their Active Directory account credentials.
- **PowerShell remote access is allowed**: Specifies that users are allowed to log on remotely to PowerShell.

There are additional predefined rights that allow access to specific applications. For example, there are predefined rights that allow users to run Performance Monitor or Server Manager without having an administrator’s password. You grant users permission to access computers by assigning them to a role that includes at least one login right. You can then give them access to specific applications or privileges using additional predefined or custom access rights.

Granting Permission to Log On

By default, zones always provide the **Windows Login** role to allow users to log on locally or remotely to computers in the zone. Users must have at least one role assignment that grants console or remote login access or they will not be allowed to access any of the computers in the zone.

> **Note**: The Windows Login role grants users the permission to log on whether they are authenticated by specifying a user name and password or by using a smart card and personal identification number (PIN).

Because the Windows Login role only allows users to log on, it is often assigned to users in a parent zone and inherited in child zones. However, the Window Login role does not override any native Windows security policies. For example, most domain users are not allowed to log on to domain controllers. Assigning

users the Windows Login role does not grant them permission to log on to the domain controllers. Similarly, if users are required to be members of a specific Windows security group, such as Server Operators or Remote Desktop Users, to log on to specific computers, the native Windows security policies take precedence.

There are additional predefined roles that grant specific rights, such as the **Rescue always permit login** role that grants users the "rescue" right to log on if audit and monitoring service is required but not available. In general, at least one user should be assigned this role to ensure an administrator can log on if the audit and monitoring service service fails or a computer becomes unstable.

Delegating Administrative Tasks in Hierarchical Zones

You can use zones to delegate administrative tasks to specific users or groups. Using hierarchical zones, you can give separate groups of administrators the authority to manage a different sets of computers and users without granting them permission to perform actions on other computers, in other zones, or on other Active Directory objects. You can also use zones to establish a separation of duties so that only specific groups or users can perform certain tasks. For example, you can create a child zone for software-development and give the dev_mgrs group authority to manage rights and roles and manage role assignments on the computers in that zone.

By creating child zones and delegating administrative tasks within those zones, you can group computers that form a natural administrative set or that should be managed by different administrative teams. For example, you might want to group computers that are managed by a local support organization in one zone and computers that are managed by a corporate IT group in another zone. You can also control what different groups of users can do within each child zone. For example, you can set up regional zones to provide a separation of duties, authorizing users in San Francisco to manage computers in their local office while a team in Barcelona has authority to join computers to the zone and manage role assignments for offices located in Spain but does not have the authority to add users or groups.

Associating Computers and Role Assignments

You can use zones to associate a set of users with a particular role assignment to a particular set of computers. This association of a group of computers with a particular role assignment is called a **computer role**. For example, you might have several computers that are dedicated to a specific function, such as hosting Oracle databases, or to a functional area, such as payroll. Some groups of users who access these computers might require a specific set of rights. For example, the database administrators who access the computers hosting Oracle databases need different rights than users who are updating payroll records in the databases being hosted.

A computer role enables you to link the privileges associated with the database administrator role assignment, such as permission to backup and restore or create new tables, with the computers that host the Oracle databases. You can configure a separate computer role for the rights required by the users processing payroll on the same set of computers. The computer role creates the link between users with a specific role assignment, database administrator or payroll department, and the computers where that role assignment applies.

If you add an Oracle database server, you add it to the computer group. If new users are assigned the database administrator role, they automatically receive the appropriate access rights on the computers hosting Oracle databases.

You can also use computer roles to specify whether you want session-level auditing for a group of computers.

Creating a New Parent Zone

In most cases, you design a basic zone structure as part of the deployment process. After the initial deployment, you can create new hierarchical zones any time you have new administrative boundaries. For example, if you acquire another organization, add offices that are managed by a different group, or restructure the organization along different functional lines, you are likely to need new zones.

What to Do Before Creating a New Parent Zone

Before you can create parent zones, you must have installed Access Manager and run the Setup Wizard. You should also have a basic zone design that describes how you are organizing information, for example, whether you are using one top-level parent zone or more than one parent zone. There are no other prerequisites for performing this task.

Rights Required for this Task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new zones, your user account must be a domain user with the following permissions:

Parent container for new zones, for example: *domain*/Delinea/Zones	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: Create Container Objects Create Organizational Unit Objects. **Note**: Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create group objects Write Description property

> **Note**: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who Should Perform this Task

A Windows domain administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How Often You Should Perform this Task

After you are fully deployed, you create new zones infrequently to address changes to your organization.

Steps for Completing this Task

The following instructions illustrate how to create a new parent zone using Access Manager. Examples of script that uses the Windows API are included in the *Software Developer's Kit* or may be available in community forums on the Delinea website. For code examples using ADEdit, see the *ADEdit Command Reference and Scripting Guide*.

To create a new parent zone using Access Manager:

1. Open the Access Manager console.
2. In the console tree, select **Zones** and right-click, then click **Create New Zone**.
3. Type the zone name and, optionally, a longer description of the zone.

   In most cases, you should use the default parent container and container type that you created when you configured the Active Directory forest, then click **Next**.

   For zones that include Windows computers, you should always use the **default zone type**, which creates the new zone as a hierarchical zone. For Windows computers, only hierarchical zones are supported. The only reasons for changing the default other settings would be if you want to:

   - Create a zone in a new location to separate administrative activity for different groups of administrators.
   - Create a zone as an organizational unit because you want to assign a Group Policy Object to the zone.
4. In most cases, you'll want to leave the **Skip permission delegation** option deselected. If you select this option, the service does not set the security descriptor for the zone; you'll need to go in and set that attribute yourself. Some organizations prefer to set security descriptors manually. Security descriptors include security information such as the object owner, who has access rights to the object, and so forth.
5. Review information about the zone you are creating, then click **Finish**.

What to Do Next

After you create a new parent zone, you might want to create its child zones.

Where you can find additional information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- How zones organize access rights and roles
- Preparing to use zones

Creating Child Zones

For Windows, the primary reason for creating child zones is to inherit role definitions and role assignments from a parent zone. Less often, you might want to use a child zone to override role definitions and assignments that you have made in a parent zone. For example, if you have created a role definitions that allows a user to run a specific application with administrative privileges in a parent zone, you can use child zones to limit the scope of that right to specific subsets of computers.

What to Do Before Creating Child Zones

Before you create child zones, you must have installed Access Manager, run the Setup Wizard to create the Zones container, and created at least one parent zone. You should also have a basic zone design that describes the zone hierarchy for the child zone. There are no other prerequisites for performing this task.

Rights Required for this Task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new child zones, your user account must be a domain user with the following permissions:

Container for the parent zones, for example if the parent zone is berlin: *domain*/MyOU/Zones/berlin	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: Create Container Objects Create Organizational Unit Objects. **Note**: Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: Create group objects Write Description property These permissions are only needed if you are supporting "agentless" authentication in the new zone.

> **Note**: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who Should Perform this Task

A Windows administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How Often You Should Perform this Task

After you are fully deployed, you create new child zones infrequently to address changes to the scope of ownership and administrative tasks.

Steps for Completing this Task

The following instructions illustrate how to create a new child zone using Access Manager.

To create a new child zone using Access Manager:

1. Open the Access Manager console.
2. In the console tree, expand **Zones** and individual zones to select the parent zone for the new child zone.
3. Right-click, then click **Create Child Zone**.
4. Type the zone name and, optionally, a longer description of the zone.

   Because this is a child zone, you should use the default parent container and container type, then click **Next**.
5. In most cases, you'll want to leave the **Skip permission delegation** option deselected. If you select this option, the service does not set the security descriptor for the zone; you'll need to go in and set that attribute yourself. Some organizations prefer to set security descriptors manually. Security descriptors include security information such as the object owner, who has access rights to the object, and so forth.
6. Review information about the child zone, then click **Finish**.

Opening and Closing Zones

Because properties and objects are organized into zones, you must open a zone to work with its contents. If you open a parent zone, its child zones are also available for you to use by default. If you open a child zone, you can choose whether to open its parent zone. Once you open a zone, it stays open until you close it and you can have multiple zones and zone levels open at the same time. If you have a large number of zones, you should close any zones you aren't actively working with for better performance.

As an alternative to opening individual or parent and child zones manually, you can automatically load all zones in a forest or all zones in a specific container at startup time. If you choose to load all zones, you cannot manually close zones.

To open an individual parent or child zone:

1. Open Access Manager.
2. In the console tree, select **Zones** and right-click, then click **Open Zone**.
3. Type all or part of the name of the zone you want to open, then click **Find Now**.
4. Select the zone to open from the list of results, then click **OK**. You can use the CTRL and SHIFT keys to select multiple zones.

After you open the zones you want to work with, you should save your changes when you exit the Access Manager console, so that the open zones are displayed by default the next time you start the console.

To close an open zone:

1. Open Access Manager.
2. Expand the zone hierarchy until you can select the specific zone name you want to close
3. Right-click, then click **Close**.
4. Click **Yes** to confirm that you want to close the zone.

To load all zones automatically:

1. Open Access Manager.
2. In the console tree, select Access Manager, right-click, then click **Options**.
3. On the **Filter Settings** tab, select **Load all zones**, then select **connected forest** to automatically load all zones in the forest or click **Browse** to navigate to specific container.

   Selecting this option prevents you from opening or closing any zones manually. You should not select the Load all zones option if you want to manually open and close individual zones for performance reasons.

Changing Zone Properties

After you create a zone, you can change its zone properties at any time. For example, if you want to change the parent zone for a child zone, you can do so by modifying the child zone's properties.

To change the properties for a zone:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand the zone hierarchy until you see the zone you want to modify.
3. Select the zone, right-click, then click **Properties**.
4. On the General tab, you can view the location of the zone in Active Directory and the zone type.

   From the General tab, you can make the following changes:

   - Change the parent zone for a child zone.
   - Modify the zone description.
   - Select a specific Licenses container for the zone to use.
   - Configure the access control list of permissions for the zone.

For example, click **Browse** to find and select a new zone to use as the parent of a child zone, then click **OK** to save the new zone properties. For Windows computers, only the properties on the General tab are applicable.

Moving a Child Zone to a New Parent Zone

You can make an existing zone a child of another zone by dragging and dropping it from one zone to another or by changing the Parent zone field on the zone's Properties General tab.

If a child zone inherits role assignments from its parent zone, the console displays a warning message and prevents you from moving the zone until you have removed the role assignments. If moving the zone creates a circular hierarchy, the console prevents you from moving the zone.

Delegating Control of Administrative Tasks

If you are the creator of a parent or child zone, you can use the Access Manager console to give other users and groups permission to perform specific types of administrative tasks within each zone you create. For example, assume you have created a zone called Finance. Certain users or groups who access computers in that zone must be able to perform administrative tasks on their own without your help. You want to give them the permissions they require to accomplish specific tasks without turning over full control to anyone except your most trusted administrative staff. Using Access Manager and the Zone Delegation Wizard, you select the appropriate groups and users for the Finance zone and specify exactly what each do. For example:

- Members of the group Finance-ITStaff are allowed to perform All administrative tasks within the Finance zone. They can change zone properties, join and remove computers from the zone, define rights and roles, and assign roles to users and groups. Only your most trusted administrative staff are members of this group.
- Members of the group FinanceManagers are allowed to join and remove computers from the zone and assign roles to users and groups.
- Members of the group FinanceUsers are allowed to add users, add groups, and join computers to the zone, but perform no other tasks.
- The users jason.ellison and noah.stone have permission to remove computers from the zone.

In most cases, each zone should have at least one Active Directory group that can be delegated to perform all administrative tasks, so that members of that group can manage their own zone. You are not required to create or use a zone administrator group for every zone. However, assigning the management of each zone to a specific user or group creates a natural separation of duties for administrative tasks.

If you delegate control for individual tasks—for example, by assigning only the join computers task to one group and only the add and remove users tasks to another—you should ensure the members of each group know the tasks they are assigned.

You can delegate administrative tasks for parent zones, for child zones, and for individual computers. Because computer-level overrides are essentially single computer zones, you can assign administrative tasks to users and groups at the computer level.

To delegate which users and groups have control over the objects in a zone:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand the zone hierarchy until you see the specific zone you want to modify.
3. Select the zone, right-click, then click **Delegate Zone Control**.
4. Click **Add** to find the users, groups, or computer accounts to which you want to delegate specific tasks.
5. Select the type of account—**User**, **Group**, or **Computer**—to search for, type all or part of the account name, then click **Find Now**.
6. Select one or more accounts from the list of results, then click **OK**.
7. Repeat Step 4 through Step 6 until you are finished adding users and groups to which you want to assign the same administrative tasks, then click **Next**.
8. Select the tasks you want to delegate to the user or group, then click **Next**.

   For example, if you want all of the members of the group you selected in the previous steps to be able perform all administrative tasks for a zone, select **All**.
9. If you are delegating the task of joining computers to a zone, you can specify the scope of computers you can join to the zone; you pick a container in Active Directory to grant access to.

   If you leave the scope blank, the scope is the domain root. Be aware that the postalAddress field is used for information about joining computers to a zone; if you lookup the permissions for people you've delegated the task of joining computers to a zone, they'll have permissions to the postalAddress field for the affected computers.

10. Review your delegation settings, then click **Finish** to close the wizard.

Granting the Authority to Perform All Administrative Tasks

Only the administrator who creates a zone has full control over the zone's properties and only that administrator can delegate administrative tasks to other users. For each zone you create, you should identify at least one user or group that can be delegated to perform all administrative tasks. For example, if you have a Finance zone, you may want to create a Finance Admins group in Active Directory and then delegate **All** tasks to that group so that members of that group can manage the zone.

Although you are not required to create or use a zone administrator group for every zone, assigning the management of each zone to a specific user or group simplifies the delegation of administrative tasks.

If members of the designated administrative group must be able to create parent or child zones, they should be assigned the rights described in Creating a new parent zone and Creating child zones.

Restricting Authority to Specific Administrative Tasks

You can use the Zone Delegation Wizard to set up fine-grain control over the specific administrative tasks different sets of users or groups can perform. For example, you can choose to grant the Join Operators group permission to join computers to the zone and no other tasks. You can then specify another group is only allowed add and remove users. If you choose to use fine-grain control over specific administrative tasks, you should ensure the members of those groups know their restricted authority.

> **Note**: If you delegate administrative tasks to one or more groups that have members logged on, you should inform the group members that they should log out and log back on so that they can perform the administrative tasks assigned to the group.

Adding Windows Computers to a Zone

To use identity and privilege management features, a Windows computer must have the Agent for Windows installed, be joined to an Active Directory domain, and joined to a zone. Depending on your organization's policies, you can either allow any authenticated user with a valid domain account to join a zone or require a domain administrator account to join a zone.

If you want to have individual users deploy the Agent for Windows on their own computers and join a zone without administrative rights, you can prepare the zone in advance and let users know which zone to join. If only domain administrators are allowed to join computers to zones, you should log on to computers with the Agent for Windows installed using an account that has appropriate administrative rights and provide a password.

Preparing Windows Computer Accounts

If joining a zone is restricted to privileged users, you may want to prepare a computer account in the zone before joining. By preparing the computer account before joining, users can add their computers to the zone without any special rights or permissions in Active Directory.

To prepare a Windows computer account using Access Manager:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand the parent and child zone hierarchy until you see the specific zone to which you want to add the computer account.
3. Right-click, then click **Prepare Windows Computer**.
4. Click **Find Now** to search for and select the computer account to add to the selected zone.
5. Click **OK** to add the computer account to the Access Manager console in the zone's Computers container.
6. A dialog box displays that asks if you want to skip permission delegation when creating the computer. In most cases, click **No**.

   If you click Yes, the service does not set the security descriptor for the zone; you'll need to go in and set that attribute yourself. Some organizations prefer to set security descriptors manually. Security descriptors include security information such as the object owner, who has access rights to the object, and so forth.

Changing the Zone for the Computer

You can move computer accounts from one zone to another at any time, if needed. Users who have administrative privileges can change the current zone on

their local computer using the agent configuration panel. You can also change the zone information for a computer from Access Manager by changing its Active Directory properties or by dragging and dropping the computer from its current to a new zone.

To change the zone for a computer using Access Manager and Active Directory properties:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand the zone hierarchy until you see the specific zone you want to modify.
3. Expand **Computers** to display the list of computers in the zone.
4. Select the computer that you want to modify, then right-click and select **AD Properties**.
5. Click the **Windows Profile** tab.
6. Click **Browse** and type all or part of the zone name, then click **Find Now**.
7. Select the new zone for the computer from the list of results, then click **OK**.
8. If the computer has role assignments defined, Access Manager prevents you from moving the computer until you remove the role assignments.

Leaving a Zone

You can remove a computer from a zone at any time. Users who have administrative privileges can leave the current zone on their local computer using the agent configuration panel. You can also remove the zone information for a computer from Access Manager by deleting the computer from its current zone. Leaving the zone does not remove the computer object from Active Directory.

To remove a computer from a zone using Access Manager:

- Open Access Manager.
- Expand **Zones** to display the list of zones, then expand the zone hierarchy until you see the specific zone you want to modify.
- Expand **Computers** to display the list of computers in the zone.
- Select the computer that you want to remove from the zone, right-click, then select **Delete**.
- Click **Yes** to confirm the removal of the computer from the zone.

Renaming a Zone

You can rename a zone at any time. For example, if your organization changes how business units are aligned, moves to a new location, or merges with another organization, you might want to update zone names and descriptions to reflect these changes. You might also want to rename zones if your initial deployment did not use a naming convention for new zones, and you want to implement one after you have agents deployed.

What to Do Before Renaming a Zone

Before you rename zones, you might want to define and document a naming convention to use for future zones or the reasons for changing the zone name. You should also identify the computers in the zone to be renamed. You do not need to restart the agent on Windows computers for the new zone name to be recognized. However, you might need to perform other administrative tasks—such as changing role assignments—after renaming a zone. There are no other prerequisites for performing this task.

Rights Required for this Task

To rename a zone, your user account must be set with the following permissions:

Parent container for an individual zone For example, a ZoneName container object, such as: *domain*/Zones/arcade	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: Write Description Write name Write Name These are the minimum permissions required to rename a zone and not allow a user or group to modify any other zone properties. You can set permissions manually, or automatically grant these and other permissions to specific users or groups by selecting the **Change zone properties** task in the Zone Delegation Wizard.

Who Should Perform this Task

A Windows administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How Often You Should Perform this Task

After you are deployed, you rename zones only when you need to address organizational changes or to implement or improve the naming conventions you use.

Steps for Completing this Task

The following instructions illustrate how to rename a zone using Access Manager.

To rename a zone using Access Manager:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand any child zones in the zone hierarchy until you see the specific zone you want to modify.
3. Select the zone to change, right-click, then click **Rename**.
4. Type the new name and, if needed, any changes to the zone description.

   You do not have to restart any Agents on the computers in the zone you have renamed. Computers will remain joined to the zone even after changing the zone name.
5. Users who have administrative privileges can verify the updated zone name on their local computer using the agent configuration panel.

Working Directly with Managed Computers

When you deploy a Agent on a computer, that computer has tools installed locally to allow you to manage access, troubleshoot agent operations, and view information about roles and role assignments, and auditing status.

Depending on the rights associated with the role you are using, you can use the tools on the managed computer to open new desktops, run individual applications with elevated privileges, connect to services on remote computers, join or change the zone for a computer, set the level of detail to record in log files, generate diagnostic information for the agent, and view detailed information about your own or other users' effective rights and roles.

Using the Agent Configuration

The Agent for Windows provides an agent configuration panel from which you can configure agent settings for the Privileged Access Service, Privilege Elevation Service, and Audit & Monitoring Service. If you have the appropriate privileges, you can use the agent configuration panel to select the zone for a computer to join, change the current zone, or remove a computer from a zone.

To use the agent configuration panel to select the zone for a local computer:

1. Log on to a computer where the Agent is deployed.
2. From the Windows Start menu, select **Agent Configuration**.
3. Click **Privilege Elevation Service**.
4. Click **Settings**.
5. On the General tab, click **Change**.
6. Click **Browse**, type all or part of the zone name, and click **Find Now** to search for the zone.
7. Select the new zone in the search results, click **OK**, then click **OK** to return General tab.
8. Click **Close** to return to the agent configuration panel.

You can also use the agent configuration panel to set logging level, view logs, and get diagnostic information about agent operations. For more information about using the agent configuration panel to configure logging and get diagnostic information, see Troubleshooting and common questions.

If you allow users to join their own computers to a zone, you should notify them of the zone to use and see that they have access to the User's Guide for Windows.

Working with Zone Role Workflow

You can enable zone role workflow in the Admin Portal so that your users can request access to systems in particular zones. Enabling zone role workflow requires having a Connector installed in the domain. For improved performance, you can also install the Client with the CSS Extension on the affected systems.

For details about how to enable zone role workflow, see Working with zone role workflow

Using Zone Role Workflow with the Connector

If you set up zone role workflow with just the Connector, be aware that there will be a delay between when the approver approves the request and when the user can access the affected systems. Although the Connector updates Active Directory immediately after the approver approves the request, a delay occurs because it can take some time to replicate the Active Directory information and also because the Agent reloads authorization information from Active Directory at specified intervals.

Using Zone Role Workflow with the Client

If you set up zone role workflow and also install the Client (so that you'll have installed both the Agent and the Client) and enable the CSS Extension on the Client, then there is no delay. Once the designated approvers approve the request, the user can access the specified system(s) immediately. The Client uses the client channel in the background to securely communicate with the Agent.

> **Note**: For deployments that have zone role workflow enabled for use with the Client, the affected systems must have Python 3.4 or later installed.

This chapter describes how to establish role-based access controls for the computers that have the Agent for Windows installed and identity and privilege management features enabled.

Basics of Authorization and Access Rights

You can use Access Manager to centrally manage what users can do on computers that have the Agent for Windows installed. For example, you can control who can log on or connect remotely for each computer in a zone through the assignment of roles. As discussed in Managing access rights and roles using zones, a **right** represents a specific operation that a user is allowed to perform.

System Rights Allow Users to Log On

For Windows computers, the most basic rights are the system rights that determine whether a user can log on locally, log on remotely, or both. The rights that grant users local and remote access are defined by default in the Windows Login role so that you can grant users access simply by assigning the Windows Login role and without defining any custom roles or any additional access rights. You can enable or disable these system rights in any custom role definition, but you cannot add, modify, or delete them.

In most cases, you can assign the Windows Login role to all local Windows users, all Active Directory users, or both, to allow users to log on locally or remotely. However, the system rights in the Windows Login role do not override any native Windows security policies. For example, most domain users are not allowed to log on locally on domain controllers. Depending on how your organization has configured native Windows security policies, users might need to be members of a specific Windows security group, such as Server Operators or Remote Desktop Users, to log on to specific computers locally or remotely.

If you would like to require multi-factor authentication for users or groups that use Delinea-managed Windows computers, you must assign them the **require MFA for login** role in addition to the Windows Login role as there is no system right to enable multi-factor authentication within the Windows Login role.

If you enable multi-factor authentication, users will be required to type their password and provide a second form of authentication before being able to log on. For example, you can configure an authentication profile that requires users to answer a phone call, click a link in an email message, respond to a text message, provide a one-time password (OTP) token, or answer a security question. Before defining this system right, however, you should be aware that multi-factor authentication for Delinea-managed Windows computers relies on the infrastructure provided by the Privileged Access Service.

For more information about preparing to use multi-factor authentication, see the Multi-factor Authentication Quick Start Guide.

In addition to the system rights that specify whether a user can log on locally or remotely, you can use the **Rescue rights** setting to specify that users in a particular role should always be allowed to log on to a computer. This option is intended as a "safety net" for "emergency" situations when users would normally be locked out. For example, if auditing is required for a role, but the agent is not running or has been removed, users are not allowed to log on. You can use the rescue rights option to allow selected administrative users access to computers when they would otherwise be locked out and prevented from logging on. Because this option allows unaudited activity, you should strictly limit its use.

> **Note**: If you do not explicitly set the Rescue rights option for any users, only the local administrator and the domain administrator accounts will have rescue rights. Those accounts are always allowed to log on by default.

Windows-specific Rights Can Grant Users Privileged Access

In general, you use the default Windows Login role for most users during the initial deployment to prevent disruptions in user access. You can then define custom roles to add specialized access rights to grant users additional privileges in a controlled manner.

For Windows computers, these specialized access rights are:

- **Desktop** access rights enable users to create additional working environments and run applications in that desktop with their own credentials but as a member of an Active Directory or built-in group. Users who are assigned to a role with desktop rights can switch from their default desktop to a desktop with administrator privileges without having to enter an Administrator password. With a desktop right, users can also run any application from their default desktop using a selected role and credentials without opening a new desktop.
- **Application** access rights enable users to run specific local applications as another user or as a member of an Active Directory or built-in group. Users who are assigned to a role with application rights can log on with their normal Active Directory credentials and run a specific application using a role with elevated privileges without having to enter the service account or Administrator password.
- **Network** access rights enable users to connect to a remote computer as another user or as a member of an Active Directory or built-in group to perform operations, such as start and stop services, that require administrative privileges on the remote computer. Users who are assigned to a role with network access rights can perform administrative operations on a remote server using a role with elevated privileges that only applies to the operations performed on the network computer without having to enter the service account or Administrator password. You can use zones to control who can connect and perform tasks on remote computers and what their elevated privileges allow them to do.

Combining Rights into Roles and Role Assignments

You can combine the system rights and specialized Windows rights into **role definitions** that reflect the needs of a specific job function, such as database administrator or web services administrator, or a particular task, such as troubleshooting application failures. You can then assign those roles to specific users and groups.

You can configure rights, role definitions, and role assignments in any parent or child zone. In most cases, you define rights and roles in a parent zone and make role assignments in a child zone.

Roles can be assigned to individual Active Directory users or to Active Directory groups. Therefore, you can manage how roles are applied to users completely through Active Directory group membership.

The rights from multiple role assignments accumulate, which provides great flexibility and granularity in how you define and assign rights and roles. For example, you can use the Windows Login role to control console and remote access, and define a second role with desktop access rights so that a user assigned to both roles could log in and create another desktop for accessing applications with administrative privileges. By separating login and desktop access rights into separate roles, not every user who is allowed to log on can create a desktop with administrative privileges.

Deciding Where to Define and Assign Roles

Because access rights are additive, it is important to consider where you define and assign roles to control who has administrative privileges on which computers. For example, it might seem reasonable to assign the predefined Windows Login role to all Active Directory users. Doing so, however, could grant broad permission to log on locally or remotely on computers to which you want to restrict access. If you assign that role in a parent zone, it is inherited along with any additional rights granted in child zones.

In most cases, it is appropriate to define roles in parent zones, but assign roles carefully in child zones to avoid granting access rights on computers that host administrative applications or sensitive information.

Adding Predefined Rights to a Zone

There are many predefined rights available that grant access to specific Windows applications. For example, there is a predefined Performance Monitor right that allows users to run Performance Monitor on a computer without being a local administrator or knowing an administrative password.

You can add any or all of these predefined rights to any zone so they are available to include in role definitions. Alternatively, you can add predefined rights to individual role definitions without adding them to zones. In either case, you create the predefined rights in the context of a role definition.

To create predefined rights in a zone:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define a predefined right.
3. Expand **Authorization > Role Definitions**.
4. Select a role definition, right-click, then select **Add Right**.
5. Select a type of right if you want to filter the list of rights displayed.

   For example, select Any Windows Rights or Any Windows Applications to list only Windows-specific rights.
6. Click **Create Predefined Rights**.
7. Select the specific predefined rights you want created in the zone you selected in Step 2 from the list of available rights, then click **OK**.

   By default, all of the selected predefined rights are added to the role definition in the zone. You can deselect any of the rights you don't want added to the role definition.
8. If you have selected at least one of the predefined rights as applicable for the role definition, click **OK**.

   If none of the predefined rights is applicable for the role definition, you can click **Cancel** to add the rights to the zone without adding them to the role definition.

You can click **Refresh** in Access Manager to see the predefined rights listed as Windows application rights.

Enabling Multi-factor Authentication for Windows Rights

In addition to the **require MFA for login** role, which requires users to provide both their password and a second form of authentication to log on to a Delinea-managed Windows computer, you can enable multi-factor authentication for a predefined right. When you define a desktop, application, or network access right, you can choose to enable multi-factor authentication for that right. For example, if you want to require multi-factor authentication before a user can open a privileged desktop, you would issue that user a role with a predefined desktop right that has multi-factor authentication enabled.

To enable multi-factor authentication for a right definition:

1. Right-click the predefined right after adding it to a role definition.
2. Select **Properties**.
3. Click the **Run As** tab and select **Re-authenticate current user** and **Require multifactor authentication**.

   Note: Before defining this right, you should be aware that multi-factor authentication for Delinea-managed Windows computers relies on the infrastructure provided by the Privileged Access Service.
4. Click **OK**.

Using Multi-factor Authentication When There are Selective Cross-forest Trusts

If you have domains in different forests that have a two-way selective trust relationship, any computer or user accounts that are used to log on to the remote forest must be granted the "Allowed to authenticate" right on the domain controllers in both forests to get role information.

In addition to granting the "Allowed to authenticate" right to users and to computers with the Agent for Windows installed, the right must also be granted to computers that host your connectors.

After you grant these computers and users the "Allowed to authenticate" right for the domains in both forests, users that are assigned a role with a multi-factor authentication right for privilege elevation will be able to authenticate using any of the authentication mechanisms that you have assigned to them.

If a connector is not allowed to authenticate on the remote domain controller, some multi-factor authentication mechanisms may fail to authenticate users.

For more information about preparing to use multi-factor authentication, see the Multi-factor Authentication Quick Start Guide.

Defining Desktop Access Rights

When users log on with their normal Active Directory credentials, Windows brings up the **default desktop** for the user logging on. You can define desktop rights to enable users to create additional working environments—new desktops—that run using their own credentials but with the privileges of an Active Directory or built-in group.

Users who are assigned to a role with desktop rights can switch from their default desktop to a desktop with elevated privileges to perform administrative tasks. For example, if assigned to a role that has a desktop right, a user can create a new desktop and switch to it when he needs perform administrative tasks such as install new software or stop running services on the local computer account. The user can perform these tasks without having to enter the service account or Administrator password.

Users who are assigned a role with desktop rights can also select any application on the computer, right-click, and run the application using a selected role. The difference between the desktop right and an application right is that the desktop right allows the user to run any applications using the privileged account defined in the desktop right. An application right restricts access to a specific application using the privileged account explicitly defined for that application.

Desktop rights are useful for users who frequently perform tasks that require the privileges associated with the Administrator account.

To define a desktop right:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define a desktop right.
3. Expand **Authorization > Windows Right Definitions**.
4. Select **Desktops**, right-click, then click **New Windows Desktop**.
5. On the General tab, type a name and a description for the desktop right.

Name	Type the name you want to use for this desktop right. For example, if the desktop allows a user to create a desktop using the privileges associated with a service account, you might include the security group in the name.
Description	Type a description for this desktop right. The description is optional. You can use it to provide a more detailed explanation of the privileges associated with the desktop.
Priority	Set the priority for this desktop right.

1. Click the **Run As** tab.

   You can browse for and select a specific group that will allow you to log on with your own credentials but with the elevated privileges of the specified group.

   Click **Add AD Groups** or **Add Built-in Groups** to search for and select a previously-defined or built-in group with the privileges you want to add to the logged in user's account.

   Select **No re-authentication required** to allow users to use the desktop right without any additional authentication.

   Select **Re-authenticate current user** if you want to prevent the desktop right and its privileges from being used by anyone not authorized to do so. Selecting this option also allows you to enable multi-factor authentication for the right. For more information, see Enabling multi-factor authentication for Windows rights.

   If you select the Re-authenticate current user option, users are prompted to re-enter their password to verify their identity before they are allowed to create a new desktop or switch between desktops. Forcing users to re-authenticate ensures the privileges associated with the desktop are only granted to users who have been assigned those privileges.

   If you select this Re-authenticate current user option for users who are authenticated using a smart card, users must enter a personal identification number (PIN) or a password to resume working with the desktop.

2. Click **OK** to save the desktop right.

Where Desktop Rights Apply

Desktop rights can be used on Windows servers and workstations that have a traditional Windows desktop. If the computer you are using is running Windows 8 or 8.1, or Windows Server 2012 or 2012 R2, Windows does not provide access to applications natively when you switch from the default desktop to a privileged desktop due to changes to the underlying interfaces and supported features within the operating system. To enable access to applications on computers running these versions of Windows, the Agent for Windows provides a custom start menu. The start menu allows you to open and run applications as you would on Windows 7 or Windows Server 2008 R2. The start menu is installed on the left side of the taskbar and displays the Delinea logo. This start menu is only available if you are using a role with Delinea desktop rights and cannot be modified.

Defining Application Rights

Application rights allow users to run specific applications using either another user account or using their own credentials but with the privileges of an Active Directory or built-in group.

When you create an application right, you specify one or more application executable files to which you want to control access. The capability to specify more than one executable file in a single application right takes into account situations in which one application might reside in different locations on different computers. For example, the executable file for SQL Server Management Studio resides in different locations in Windows 2005, Windows 2008, and Windows 2012. By specifying all instances of the executable file in one application right, you can use that application right to control access to SQL Server Management Studio on computers running any of those operating systems.

You can also use Delinea application utilities to allow access to common administrative tasks such as software installation, network, and Windows feature management. For more information on using these utilities, see Using Delinea application utility rights

> **Note**: Although it is possible to define different applications (for example, SQL Server Management Studio and Internet Explorer) in one application right, this is not a recommended practice. Instead, it is recommended that you create separate application rights for different applications.

How to Specify Which Applications are in an Application Right

You can specify which application executable files are in an application right in these ways:

- You can specify the path and file name of an application executable file. You can perform this operation in two ways:
  - Manually, by typing or pasting the path and file name into an application right definition form. Specifying files manually is recommended only if you need to include a small number of files in the definition—typically just one or two. See Defining an application right manually for more information.
  - By navigating to the executable file or a running process that was launched by the executable file. After locating the executable file, you can import the path and file name into the application right definition form. See Using an installed application or running process to create application rights for more information.
- You can specify search criteria for application executable files, and then include all application executable files that match those criteria in the application right. You can perform this operation in two ways:
  - Manually, by typing or pasting values into search criteria fields. See Defining an application right manually for more information.
  - By importing values into search criteria fields from an executable file or from a running process that was launched by the executable file. See Using an installed application or running process to create application rights for more information.

See Examples of application right definitions for examples of defining application rights in all of these ways.

Defining an Application Right Manually

This section describes how to create an application right by manually typing or pasting information into several application right definition forms.

> **Note**: Alternatively, you can import information into application right definition forms from an executable file or from a running process that was launched by the executable file. See Using an installed application or running process to create application rights for more information.

To define an application right manually:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define an application right.
3. Expand **Authorization > Windows Right Definitions**.
4. Select **Applications**, right-click, then click **New Windows Application**.
5. On the General tab, type a name and a description for the application right, and specify a priority for the application right.

Name	Type the name you want to use for this application right. For example, if the right allows a user to run SQL Server Configuration Manager using the privileges associated with a security group, you might include the service account in the name. For example, you might use a name like SQL Config Manager.
Description	Type a description for this application right. The description is optional. You can use it to provide a more detailed explanation of the privileges associated with running the application.
	Set the priority for this application right. If more than one application right is added to the same role definition, the priority value determines the application right to use when users assigned to that role open that application. The lower the value, the higher the priority. For example, a right with the priority of 1 takes precedence over a priority value of 2. If the application rights have the same priority value, the application right listed first under the role definition is used.

1. Click the **Match Criteria** tab and use it to create or edit application definitions. Each application definition specifies one application or a group of applications. The set of application definitions displayed in the **Match Criteria** tab defines the set of applications that can be run by this application right.

   In the **Match Criteria** tab, click **Add** to create a new application definition.

   The Definition Settings dialog appears.

2. In the upper portion of the Definition Settings dialog, provide this information about the application definition.

	Type a description for this application definition. For example, if the definition specifies one executable file (such as SQL Server Management

Description	Studio for Windows 2005), you might type **Windows 2005 SQL Server Management Studio** here. Or, if the definition specifies more general criteria so that multiple executable files (such as SQL Server Management Studio for all versions of Window) can run, you might type a more general description such as **SQL Server Management Studio**.
File Type	Select the type of executable file for this definition. If you are constructing the definition so that it specifies multiple executable files, all files must all be of the type that you specify here. Supported file types are: .bat .cmd .com .cpl .exe .msc .msi .msp .ps1 .vbs .wsf

1. To specify executable files in this definition by typing or pasting the file name and location, select the **Path** option. Go to Step 9 and continue from there.

   Specifying files in this way is recommended only if you need to include a small number of files in the definition—typically just one or two.

   To specify a larger number of executable files in this definition, it is recommended that you select file parameters that are common to the set of files. Files that match the parameters are then included in the definition. To do this, go to Step 10 and continue from there.

2. Perform this step to specify a small number of executable files in this definition. In this step, you type or paste information about the executable file name, location(s), and arguments. When you are done with this step, go to Step 11 and continue from there.

Name	Type the name of the application executable file. If this field is defined, you must also select a path option (standard system path or a specified path). For example, to specify the SQL Server Management Studio executable, type Ssms.exe.
Standard system path	Select **Standard system path** to use the directories where the user would normally find the application specified. For example, to use the application executable in its default directory, select **Standard system path**.
Specify path	Select **Specify path** if you want to define the location of the application specified. If you select this option, you can specify one or more paths, separated by a semicolon (;). Supported path variables are %systemroot%, %system32%, %syswow64%, %program files%, %winagentinstall%, and %program files(x86)% (note that a space between "program" and "files" is required). For example, to specify the location of the SQL Server Management Studio executable file in Windows 2008, type C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE.
Arguments	If you selected a file type of .msc in Step 7, the **Arguments** option is required. The **Arguments** option is optional for all other file types. Select the **Arguments** option and leave the argument field blank to specify that the application cannot accept any arguments. To specify that the application can run using any argument, leave the Arguments option deselected. For example, if you specified the SQL Server Management Studio executable and left the **Arguments** option deselected, users can run SQL Server Management Studio with any option on a local computer with elevated privileges. If you want to restrict the arguments allowed, in the argument field type the list of arguments to allow. Valid arguments be must enclosed by quotation marks and separated by a space. For example, to allow users to run the specified application using argument1, argument2, or argument3, you would specify the list of arguments like this: "argument1" "argument2" "argument3" By default, arguments that you specify do not need to be a case-sensitive match, but do need to be an exact match (that is, a match is returned if the actual argument is a partial match of the argument string that you specify). If arguments must be a case-sensitive match for a particular application, select the **Keep arguments case sensitive** option. If arguments can be a partial match for a particular application, deselect the **Match whole string only** option.

1. Perform this step to specify a larger number of executable files in this definition. In this step, you use the **File details** area to specify characteristics that are used to search for applications to include in this definition. All of the characteristics that you specify must be met in order for an application to be a match. For example, if you specify a product name of Microsoft SQL Server and a company name of Microsoft Corporation, all executable files that meet both of those criteria are included in this definition.

   Note: This step describes how to manually fill in each field in the **File details** area. You can select any combination of these fields to specify the file characteristics for which to search. Alternatively, you can populate fields in the Definition Settings dialog by importing values from an installed executable file or from a running process. Filling in fields by importing is faster and more accurate than filling in fields manually one at a time. For details about filling in fields by importing, see Using an installed application or running process to create application rights.

Product Name	Select an operator (is or contains) from the drop-down list and in the provided field type the product name for which to search. If you select is, matches are returned for product names that exactly match the string that you type here. If you select contains, matches are returned for product names that contain the string that you type here anywhere in the product name.
Company	Select an operator (is or contains) from the drop-down list and in the provided field type a company name for which to search.
File Description	Select an operator (is or contains) from the drop-down list and in the provided field type a file description for which to search.
Volume Serial #	Select an operator (is, contains, starts with, or ends with) from the drop-down list and in the provided field type a serial number for which to search. The supported format is 8-character hex string (FFFFFFFF). This criterion is matched only if the executable file was from CD/DVD media.
Publisher	Select an operator (is, contains, starts with, or ends with) from the drop-down list and in the provided field type publisher information for which to search. For example, publisher information could look similar to: CN=Acme Corporation, OU=Digital ID Class 3 - Microsoft Software Validation v2, O=Acme Corporation, L=Sunnyvale
Product Version	Select an operator (equal, earlier or equal, or later or equal) from the drop-down list and in the provided field type product version information for which to search. For example, the product version could look similar to: 3.1
File Version	Select an operator (equal, earlier or equal, or later or equal) from the drop-down list and in the provided field type file version information for which to search. For example, the file version could look similar to: 3.1.2
File Hash	Select this option to match applications using the encrypted file hash for the application. The file hash for the application is generated using the SHA-1 encryption algorithm, which is FIPScompliant. You can click **Import Process** or **Import File** and select an application to populate the File Hash field for which to search. Only applications with a hash string that is exactly the same as the string generated by the MD5 algorithm are matched. You can only use file hash matching to identify an application for files that are less than 500MB to limit the CPU and memory used to calculate the file hash. If the file with matching hash information is larger than 500MB, an empty value is returned for the file hash field.
Owner	In the provided field, type owner information for which to search. Matches are returned for owner information that exactly matches the string that you type here. Owner information can be: AD user/group/builtin (SID) local user (user name) local group (group name) For example, the owner could look similar to: NT AUTHORITY\SYSTEM DEMO\Ed.Admin (this is an AD user account) Amy Adams (this is a local user account)

1. Optionally select the **Application requires administrative user** option to specify that applications in this definition run only if RequestedExecutionLevel is set to requireAdministrator in the application manifest. If you select this option, the applications in this definition run only for administrators and require that the applications be launched with the full access token of an administrator. This option applies only to .exe files.
2. Click **OK** to save the definition. You are returned to the **Match Criteria** tab, and the new or modified definition appears in the **Match Criteria** list of definitions.
3. Click the **Run As** tab and select the account that has the privileges you want to enable for this application right.

   You can browse for and select a specific user account or have the application run using the logged in user's account credentials but with the elevated privileges of a specified group. Click **Add AD Groups** or **Add Built-in Groups** to search for and select a previously-defined or Built-in group with the privileges you want to add to the logged in user's account.

   In most cases, you select a specific user account only if the application should run as a service account. However, some applications require a specific privileged user account to be used. For example, Microsoft System Center Operations Manager (SCOM) and Exchange require a user account. If you are defining an application right for an application that requires a privileged user account rather than membership in a privileged group, you should create a service account and use that account for the run-as account.

   Select **Re-authenticate current user** if you want to prevent the application right and its privileges from being used by anyone not authorized to do so. Selecting this option also allows you to enable multi-factor authentication for the right. For more information see Enabling multi-factor authentication for Windows rights.

   If you select this option, users are prompted to re-enter their password to verify their identity before they are allowed to select a role for running a local application. Forcing users to reauthenticate ensures the privileges associated with the application right are only granted to users who have been assigned those privileges.

If you select this option for users who are authenticated using a smart card, users must enter a personal identification number (PIN) or a password to resume working with the application.

4. Click **OK** to save the application right.

Using Application Utility Rights

This section describes how you can manage user access to Windows programs and features using application utility rights.

There are many common administrative tasks such as managing software installations, changing network settings, and adding or removing Windows features that require access to the explorer.exe application on Windows systems. Because granting users privileged access to explorer.exe can allow the user to perform many other tasks that you may want to remain restricted, you can use the Delinea application utilities, **Application Manager**, **Network Manager** and **Windows Feature Manager**, to grant access to these tasks using the corresponding predefined rights.

When you create a new zone, the Delinea utility rights are automatically added to the list of Windows Right Definitions. However, in zones that existed before the addition of these utility rights, you may need to add them by following the procedure below.

To add the Delinea Utilities to the list of Windows Right Definitions

1. Right click **Windows Right Definitions** and select **Add predefined rights**.

   Windows Right Definitions can be found in the following location:

   The application rights can be found in the following location:

   Access Manager > Zones > Zone Name > Authorization > Windows Right Definitions

2. Select the rights you would like to add and click **OK**.

   The rights will now appear under **Applications**.

It is important to note that if you do not install the Agent for Windows in the default location during the installation or upgrade process, users who are assigned these rights may not be able to access the corresponding applications. If you have installed the agent in a location other than the default location, you can specify a variable in the application right settings to allow them to be used by assigned users by doing the following:

To specify the application right path

1. Right click on the application right and select **Properties**.

   The application rights can be found in the following location:

   Access Manager > Zones > Zone Name > Authorization > Windows Right Definitions > Applications.

2. Click the **Match Criteria tab**, and then click **Edit.**
3. Check the **Path** box in the **Commands components** section, and select **Specific path**.
4. In the **Specific path** field, enter the following variable: %winagentinstall%

Do this for each of the Utility application rights.

Application Manager

Application Manager is a utility that allows a user to manage installed software. Application Manager is similar to the Windows utility Programs and Features. It can allow users who are assigned a role with the **Utility - Application Manager** right to Refresh, Uninstall, Change, or Repair installed software.

Windows Feature Manager

When you assign workstation users a role with the predefined right **Utility - Windows Feature Manager**, they will be able to access the normal Windows Feature Manager, where they can choose what Windows features to add or remove.

When you assign server users a role with this right, the Windows Feature Manager will launch. This utility is similar to the normal Windows utility, with a few notable differences.

Opening the utility will launch a wizard. When you select whether to add or remove roles and features on the first screen of the wizard, you can only perform one action at a time. For example, if you choose **Add roles and features**, you will not be able remove any installed features until you go back to the initial screen and choose **Remove roles and features**.

Additionally, when you attempt to install features that require the installation of dependent components, you will be prompted to add those features. All features with one or more components installed will appear with a check mark next to the name.

Network Manager

When you assign users a role with the predefined right Utility - Network Manager, they will be able to access the Delinea version of Network Manager that is similar to the Windows version.

Users assigned a role with this right can view a list of network adapters for Ethernet and wireless connections and configure their IP and DNS settings.

Using an Installed Application or Running Process to Create Application Rights

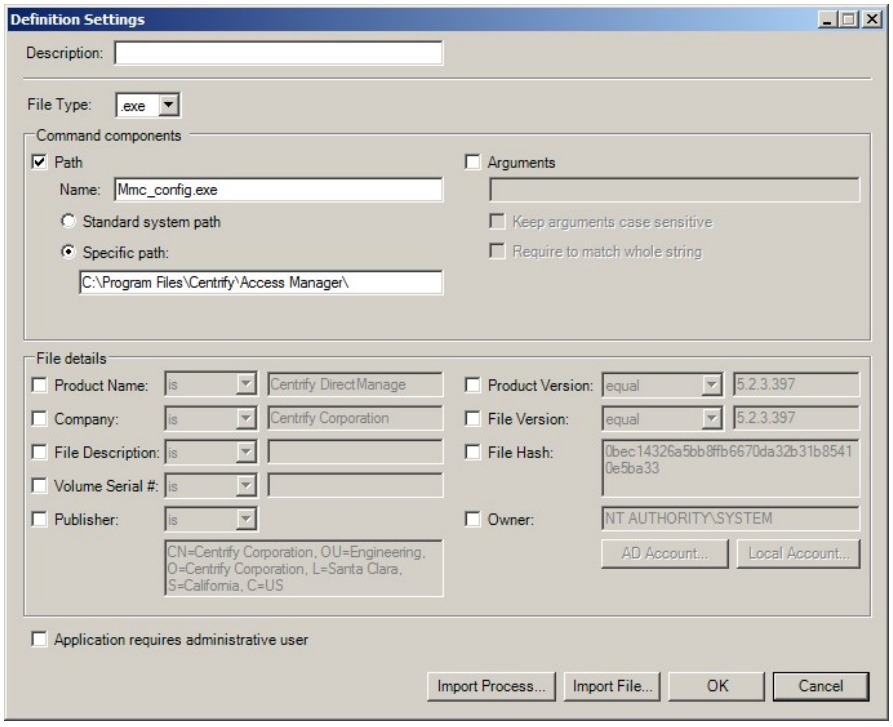
This section describes how to create an application right by importing values from an installed executable file or from a running process. After values are imported into the application right definition form, you can select which fields to use as search criteria for matching applications. Applications that match the search criteria are included in the application definition.

For more information about filling in fields by importing, see Examples of application right definitions.

To define an application right based on an installed application:

1. Follow the procedure for creating a new application right manually to the point where the Definition Settings dialog opens (see Defining an application right manually).
2. In the Definition Settings dialog, click **Import File**.
3. Navigate to an application executable file, highlight the file, and click **Open**.

   Fields in the Definition Settings dialog fill in with all of the information that is available for the file that you selected. For example, if you navigated to C:\Program Files\Centrify\Access Manage and selected the Mmc_config.exe file, the Definition Settings dialog would look similar to this:



   Notice that:

- The **File Type** field is set to .exe.
- The **Path** option is selected, and the file name and path name are filled in.
- Most fields in the **File details** section are filled in, but none are selected.

  The settings shown in this example specify that only the Mmc_config.exe <!---TODO update path ---> file located in C:\Program Files\Centrify\Access Manage is included in the application right. The information in the **File details** section is not used because no options in that section have been selected.

4. Choose whether to expand the definition to include other executable files, or to save the definition as it is currently defined (so that it specifies only the Mmc_config.exe file shown here).

   To expand the definition to include other executable files, go to Step 5 and continue from there.

   To save the definition as it is currently defined:

   - In the **Description** field, type a description for this application definition. This is the string that displays in the list of application definitions on the **Match Criteria** tab.
   - Click **OK**.
   - Continue to define the application right as described in Defining an application right manually.

5. To expand the definition to include other executable files, use the **File details** area to specify characteristics that are used to search for executable files. All of the characteristics that you specify must be met in order for an executable file to be a match. See Defining an application right manually for details about operators and syntax for each option in the **File details** area.

   - Deselect the **Path** option.

     This step is necessary because all of the search options that you select use the AND operator when the search executes. If you leave the **Path** option selected, the search is constrained to this location and the definition will include only the file that is specified in the **Name** field.

   - In the **File details** area, select options to define search criteria for executable files.

     Selecting criteria that are more general will usually result in a greater number of executable files being included in the definition. In the example shown in Step 3, you would select only the **Company** option if you wanted to allow this definition to run all .exe files having a company name tag of Acme Corporation. Select additional options to limit the scope of the search so that fewer executable files are included in the definition.

   - In the **Description** field, type a description for this application definition. This is the string that displays in the list of application definitions on the **Match Criteria** tab.
   - Click **OK**.
   - Continue to define the application right as described in Defining an application right manually. When you are done, the application right is available to use.

To define an application right based on a running process:

1. Follow the procedure for creating a new application right manually to the point where the Definition Settings dialog opens (see Defining an application right manually).
2. In the Definition Settings dialog, click **Import Process**.

   A list of running processes displays. By default, the list does not include these processes:

   Processes having an owner of SYSTEM, Local Service, or Network Service

   - conhost.exe
   - dllhost.exe
   - dwm.exe
   - explorer.exe

- svchost.exe
- taskhost.exe

  To display these processes, select the **Show all processes** option.

  Note: System Idle Process and processes having unsupported file extensions (for example, .scr) are never shown.

3. Highlight a process and click **OK**.

   Fields in the Definition Settings dialog fill in with information from the executable file that launched the process that you selected.

4. Select executable files to include in this definition as described in Step 4 on page 149 through Step 5 on page 150. When you are done, the application right is available to use.

Examples of Application Right Definitions

This section contains these examples of how to use the Definition Settings dialog to specify an application right definition:

- Example 1: Manually specify one application path and file name—Describes how to define an application right to run the Access Manager console by manually entering the path name and application name.
- Example 2: Manually specify one application residing in two locations—Describes how to define an application right to run SQL Server Management Studio on Windows 2008 and Windows 2012 systems by manually entering the application name and the path names to the application on both systems.
- Example 3: Specify one application by importing its location—Describes how to define an application right to run the Access Manager console by <!---TODO update filename---> navigating to the centrifydc.msc file and importing its information.
- Example 4: Specify several applications by importing and specifying search criteria—Describes how to define an application right to run SQL Server Management Studio on several versions of the Windows operating system by navigating to the Ssms.exe file on Windows 2008, importing its information, and constructing application search criteria based on that information.

Example 1: Manually specify one application path and file name

In this example, it is assumed that you want to create an application right to run the Access Manager console application, and you know the path and file name of the application executable file.

1. Open the Definition Settings dialog and fill it in as follows:

   Description—Type a name of your choice (for example, **Default Access Manager Console Application**).

   Path—Select this check box.

   Name—Type the application name; in this case <!---TODO update filename---> centrifydc.msc.

   Arguments—Select this check box and specify which arguments can be executed through this application right.

   Specific path—Select this option and type the full path name to the <!---TODO update filename---> **centrifydc.msc** executable file: <!---TODO update path--->

   C:\Program Files\Centrify\Access Manager

2. Click **OK** to save the application right definition setting.

Example 2: Manually specify one application residing in two locations

In this example, it is assumed that you want to create an application right to run SQL Server Management Studio on Windows 2008 and Windows 2012 systems. The SQL Server Management Studio executable file resides in different locations in those operating systems, and you know the paths those locations.

1. Open the Definition Settings dialog and fill it in as follows:

   Description—Type a name of your choice (for example, **SQL Server Management Studio 2008/2012**).

   Path—Select this check box.

   Name—Type the application name; in this case Ssms.exe.

Arguments—Optionally select this check box and specify which arguments can be executed through this application right.

Specific path—Select this option and type the full path names to the **Ssms.exe** executable file in Windows 2008 and Windows 2012. Separate the path names with a semicolon(;)

C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\ManagementStudio

2. Click **OK** to save the application right definition setting.

Example 3: Specify one application by importing its location

This example is similar to Example 1; it is assumed that you want to create an application right to run the Access Manager console application. Unlike in Example 1, you are not sure of the path name to the application executable file and you will navigate to it rather than type it in the form.

1. Open the Definition Settings dialog.
2. Click **Import File**.
3. Navigate to the <!---TODO update filename---> centrifydc.msc executable file, highlight it, and click **Open**.
4. Verify that the Definition Settings dialog fills in with application information.
5. In the Description field, type a name of your choice (for example, Default Access Manager Console Application).
6. Click **OK** to save the application right definition setting.

Example 4: Specify several applications by importing and specifying search criteria

This example is similar to Example 2; it is assumed that you want to create an application right to run SQL Server Management Studio on more than one version of the Windows operating system, starting with Windows 2008. Unlike in Example 2, you do not want to constrain the latest version of Windows to Windows 2012. Instead, you want to account for future versions of Windows and provide the capability to run SQL Server Management Studio on future Windows releases.

1. Open the Definition Settings dialog on a Windows 2008 system.
2. Click **Import File**.
3. Navigate to the Ssms.exe executable file, highlight it, and click **Open**.

   The Definition Settings dialog fills in with information from the Windows 2008 version of Ssms.exe.
4. Deselect the **Path** option so that the definition is not constrained just to that location.
5. Select the **File Description** option and keep the default operator and string.
6. Select the **Product Version** option and change the operator from **equal** to **later or equal**.

   The definition is now configured to include all .exe files having a file description tag of **SSMS - SQL Server Management Studio** and a product version later than or equal to the version that is installed on this Windows 2008 system.
7. In the Description field, either keep the string that was imported with the Ssms.exe file or type a description of your choice.
8. Click **OK** to save the application right definition setting.

Defining Network Access Rights

Network access rights allow users to access services on remote computers using another user account on the remote computer. Users who are assigned to a role with network access rights are only granted the elevated privileges when accessing the remote computer.

To define a network access right:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define an application right.
3. Expand **Authorization > Windows Right Definitions**.
4. Select **Network Access**, right-click, then click **New Network Access**.
5. On the General tab, type a name and a description for the network access right.

Name	Type the name you want to use for this network access right. For example, if the right allows a user to connect remotely to a Microsoft SQL Server instance using the privileges associated with a database system administrator account, you might include the SQL login name. For example, you might use a name like sysadmin.
Description	Type a description for this network access right. The description is optional. You can use it to provide a more detailed explanation of the privileges associated with this right.
Priority	Set the priority for this application right. If more than one network access right is included in the roles selected, the priority value determines which network access right to use. The lower the value, the higher the priority. For example, a right with the priority of 1 takes precedence over a priority value of 2. If users have multiple roles selected, the priority value of the network access right determines which network access right takes precedence over the access rights in other roles. For more information about selecting multiple roles for connecting to remote servers, see Scenario: Using multiple roles for network resources.

1. Click the **Access** tab to select the account that has the privileges you want to enable for accessing the remote computer.

   You can browse for and select a specific user account, create a new account, or access the remote computer using the logged-in user's account credentials but with the elevated privileges of a specified group account. Click **Add AD Groups** or **Add Built-in Groups** to search for and select a previously-defined or Built-in group with the privileges you want to add to the logged in user's account.

   In most cases, you select a specific user account only if accessing the remote computer using a service account.

   Select **Re-authenticate current user** if you want to prevent the network access right and its privileges from being used by anyone not authorized to do so. Selecting this option also allows you to enable multi-factor authentication for the right. For more information see Enabling multi-factor authentication for Windows rights.

   If you select this option, users are prompted to re-enter their password to verify their identity before they are allowed to select a role for accessing applications on a remote computer. Forcing users to reauthenticate ensures the privileges associated with the network access right are only granted to users who have been assigned those privileges.

   If you select this option for users who are authenticated using a smart card, users must enter a personal identification number (PIN) or a password to resume working with the remote server.

2. Click **OK** to save the network access right.

Using Network Access Rights When There are Two-way Selective Cross-forest Trusts

If you have domains in different forests that have a selective two-way trust relationship, any computer or user accounts that are used to log on to the remote forest must be granted the "Allowed to authenticate" right on the domain controllers in both forests to get role information. After you grant the computer used to access the remote server the "Allowed to authenticate" right for the domains in both forests, you can select roles that grant network access rights from either forest.

If an account is not allowed to authenticate on the remote domain controller, you cannot view or select roles that would otherwise allow you to perform tasks on the remote server.

Defining Custom Roles with Specific Rights

Rights can be combined or used independently of each other to create role definitions. Role definitions describe job functions that require a specific set of rights, including the specific days and times the role should be available for performing the operations allowed. If you have created desktop, application, or network access rights, you must create at least one role definition to use these rights.

To create a new role definition for a job function, you need to do the following:

- Create a new role and specify when the role is available.
- Specify how users in the role are allowed to log on.
- Add specialized Windows access rights to the role, as applicable.
- Specify whether the role requires multi-factor authentication before it can be selected.

In most cases, creating a separate role definition for each access right gives you the most granular control over what users assigned to a role can do. For example, if you create separate role definitions for desktop, application, and network access rights, you can choose which apply to specific users and groups

through role assignments.

Creating a Role Definition with Desktop Rights

Before you can make the desktop rights you have defined available to users or groups, you must create one or more role definitions that include those rights. Desktop rights are especially useful to include in roles for users who frequently perform tasks that require the privileges associated with the Administrator group.

To create a new role definition with desktop rights:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define a new role that includes a desktop right.
3. Expand the **Authorization** node.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a role name and optional description for the role.

   The description can include details about time restrictions for the role and whether the role is audited or not.
6. Select **Allow local accounts to be assigned to this role** if you want to be able to assign local users or groups to the role you are creating.

   If you do not select this option, only Active Directory domain users can be assigned to the role.
7. Click **Available Times** and use the grid to specify when to allow or deny access for this role definition if you want to restrict when this role is available.
8. Click the **System Rights** tab and select **Console login is allowed** to allow users in the role to log on locally.

   To use the desktop right, the user must be able to log on locally on the computer. If you want to allow users to log on using a remote desktop connection, you can also select **Remote login is allowed**.

   > **Note**: Remote computers must be configured to allow remote desktop connections for the "Remote login is allowed" right to be valid. You can configure a computer to allow remote desktop connections by right-clicking Computer and selecting Properties or from the System Control Panel, then clicking **Remote settings**.

   Users must be assigned to at least one role with either console login or remote login rights to access any computers where the Agent for Windows is installed. You can grant access using the Windows Login role definition or the system rights in any custom role definition.

   The Windows right **PowerShell remote access is allowed** allows you to log on remotely to PowerShell.

   If you want to allow users to log on even when the Windows agent isn't running or when audit and monitoring service is required but not available, you can select the rescue right. Because this right allows users to log on without having their activity audited, you should only assign roles with this right to trusted administrators or under controlled conditions. For example, assume you have a computer with sensitive information that normally requires all user activity to be audited. If that computer has application or operating system issues that require you to disable auditing temporarily, you can use a role with the rescue right to log on to that computer to diagnosis and fix the issue.
9. In the **Authentication** tab, you can add multi-factor authentication. If you want to require multi-factor authentication for users to access the role, select **Require multi-factor authentication for login**. You can also require multi-factor authentication for access to individual rights when you define the rights to add to roles. For more information see Enabling multi-factor authentication for Windows rights.
10. Click the **Audit** tab and select an option.

    If you select **Audit not requested/required**, users can log on to audited computers without having their session activity recorded. An audit trail event is recorded in the Windows event log when users open a desktop with this role, but the detailed record of what took place during the session is not captured.

    If you select **Audit if possible**, session activity is recorded when users open a desktop with elevated privileges on audited computers and not recorded when they log on to computers where audit and monitoring service is not enabled or audited computers when auditing is not currently running.

    If you select **Audit required**, users can only open a desktop with elevated privileges when auditing is running. If audit and monitoring service is not available or not currently running, the role is not available and users cannot use the elevated privileges.

11. Click **OK** to save the role definition.
12. Select the role definition, right-click, then click **Add Right** to add a desktop right to the role definition.
13. Select the desktop right from the list of rights from the current zone and from any parent zones, then click **OK** to add the right to the role definition.

Creating a Role Definition with Application Rights

Before you can make the application rights you have defined available to users or groups, you must create one or more role definitions that include those rights. Application rights are especially useful to include in roles for users who infrequently require access to specific applications with the privileges associated with the Administrator account or a service account on a local computer.

To create a new role definition with application rights:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define a new role that includes an application right.
3. Expand the **Authorization** node.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a role name and optional description for the role.

   The description can include details about time restrictions for the role and whether the role is audited or not.
6. Click **Available Times** and use the grid to specify when to allow or deny access for this role definition if you want to restrict when this role is available.
7. Click the **System Rights** tab and select **Console login is allowed** to allow users in the role to log on locally.

   To use the Run as selected role utility and an application right, the user must be able to log on locally on the computer where the application runs. If you want to allow users to log on using a remote desktop connection, you can also select **Remote login is allowed**.

   Users must be assigned to at least one role with either console login or remote login rights to access any computers where the Agent for Windows is installed. You can grant access using the Windows Login role definition or the system rights in any custom role definition.

   If you want to require multi-factor authentication for users to access the role, select **Require multi-factor authentication**. You can also require multi-factor authentication for access to individual rights when you define the rights to add to roles. For more information see Enabling multi-factor authentication for Windows rights.
8. Click the **Audit** tab and select an audit and monitoring service option.
   - If you select **Audit not requested/required**, users can log on to audited computers without having their session activity recorded. An audit trail event is recorded in the Windows event log when users select this role to run the application, but the detailed record of what took place during the session is not captured.
   - If you select **Audit if possible**, session activity is recorded when users select this role to run the application and not recorded when they use the application on computers where audit and monitoring service is not enabled or audited computers when audit and monitoring service is not currently running.
   - If you select **Audit required**, users can only select this role to run the application when audit and monitoring service is running. If audit and monitoring service is not available or not currently running, the role is not available and users cannot use their elevated privileges.
9. Click **OK** to save the role definition.
10. Select the role definition, right-click, then click **Add Right** to add the application right to the role definition.
11. Select the application right from the list of rights from the current zone and from any parent zones, then click **OK** to add the right to the role definition.

Creating a Role Definition for Network Access Rights

Before you can make the network access rights you have defined available to users or groups, you must create one or more role definitions that include those rights. Network access rights are especially useful to include in roles for users who require remote access to network services with the privileges associated with the domain Administrator account or a service account on the remote computer.

1. Open the Access Manager console.

2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to define a new role that includes a network access right.
3. Expand the **Authorization** node.
4. Select **Role Definitions**, right-click, then click **Add Role**.
5. Type a role name and optional description for the role.

   The description can include details about time restrictions for the role and whether the role is audited or not.
6. Click **Available Times** and use the grid to specify when to allow or deny access for this role definition if you want to restrict when this role is available.
7. Click the **System Rights** tab and select **Remote login is allowed** to allow users in the role to connect to services on the remote computer.

   The user must be able to connect to the computer remotely to perform administrative tasks on that computer. If you want to allow users to log on locally, you can also select **Console login is allowed**.

   Users must be assigned to at least one role with either console login or remote login rights to access any computers where the Agent for Windows is installed. You can grant access using the Windows Login role definition or the system rights in any custom role definition.

   If you want to require multi-factor authentication for users to access the role, select **Require multi-factor authentication**. You can also require multi-factor authentication for access to individual rights when you define the rights to add to roles. For more information see Enabling multi-factor authentication for Windows rights.
8. Click the **Audit** tab and select an auditing option.

   If you select **Audit not requested/required**, users can connect to remote audited computers without having their session activity recorded. An audit trail event is recorded in the Windows event log when users select this role to connect to remote servers, but the detailed record of what took place during the session is not captured.

   If you select **Audit if possible**, session activity recorded when users log on to audited computers and not recorded when they log on to computers where audit and monitoring service is not enabled or audited computers when audit and monitoring service is not currently running.

   If you select **Audit required**, users can only log on to audited computers when audit and monitoring service is running. If audit and monitoring service is not available or not currently running, the role is not available and users cannot use their elevated privileges.
9. Click **OK** to save the role definition.
10. Select the role definition, right-click, then click **Add Right** to add a network access right to the role definition.
11. Select the network access right from the list of rights from the current zone and from any parent zones, then click **OK** to add the right to the role definition.

Combining Rights in the Same Role Definition

The previous sections illustrate how to create custom role definitions specifically for desktop, application, or network access rights. You can also combine multiple rights in the same role definition. For example, you can create a role definition that allows a user to open a specific application on the local computer using a service account with elevated privileges. The same role definition can also include a network access right that enables the user to modify information on a remote server.

Assigning Users and Groups to a Role

You can assign a role to an Active Directory user or to an Active Directory group. You can assign a role that is defined in the current zone or in a parent zone. You can also specify optional start and end times for the role assignment.

To assign users and groups to a role in a zone:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to make role assignments.
3. Expand **Authorization**.
4. Select **Role Assignments**, right-click, then click **Assign Role**.

5. Select the role definition from the list of roles, then click **OK**.

   By default, the role is set to start immediately and never expire. You can set a **Start time**, **End time**, or both start and end times for the role assignment. For example, if the role applies to a contractor who will be hired for a specific amount of time and you want to automatically disable the role after they finish the job and leave the organization, you can specify the start and end times when you assign the role.

6. Select whether the role assignment applies to all Active Directory accounts, all local accounts, or specific Active Directory and local accounts.

   To assign the role to specific accounts, click **Add AD Account** to search for and select the Active Directory groups or users to assign to the role, then click **OK**.

Rights and Role Assignments for Local Users

The rights you assign to users and group in a particular role apply to Active Directory users and groups. They can also apply to locally-defined users and groups if you configure the role definition to allow local accounts to be assigned to the role. All Windows users, including local users, must be assigned at least one role that allows them log on locally, remotely, or both.

Restricting Roles that Include Network Access Rights

Because role definitions can include a combination of rights and you can assign roles to local users, Active Directory users, or both, it is possible for you to assign roles that include network access rights to local accounts. Access Manager does not prevent you from configuring role definitions or role assignments in this way. However, users who log on with a local account will not be allowed to select the Advanced View or those network access rights for the remote computer. Therefore, you should avoid configuring role definitions that include network access rights and allow local accounts. Instead, you should keep role definitions that include network access rights separate from role definitions that allow local accounts to be assigned.

Making Rights and Roles Available in Other Zones

The access rights and role definitions that you create are specific to the zone where you configure them, and to any child zones of that zone. Once configured, though, you can copy and paste or drag and drop the definitions from one zone to another. After you import the information into a new zone, you can modify any of the details you have previously defined. For example, you can choose to export all the rights you have defined in one zone but create a completely new set of role definitions for those rights in the import zone.

Rights, roles, and role assignments are all inherited from parent to child zones, so generally there is no need to import or export roles within a zone hierarchy, but you may want to do so across zones. For example, if you have set up separate parent zones for different lines of business or different functional groups in your organization, you might want to import rights and roles from one business unit or functional group to another.

Exporting a Zone's Rights and Role Definitions

You can export right and role definitions to an xml file that you can then use to import these definitions into another zone.

To export rights and role definitions:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone that has the rights and roles you want to export.
3. Expand Select the **Authorization** node, right-click, then click **Export Roles and Rights**.
4. Select the information you want to export, then click **Next**.
5. Click **Browse** to specify a location and file name for the export file, then click **Next**.
6. Review the information to be exported, then click **Finish**.

Importing Rights and Role Definitions into a New Zone

You can import rights and role definitions that you have previously saved from a different zone. You can also copy a paste or drag and drop rights and roles to a different zone.

To import rights, role definitions, and role assignments:

Before you begin, be certain you have saved rights and role definitions from a different zone and know the location of the xml file in which they are saved.

1. Open Access Manager.
2. Expand **Zones** and the parent zone or child zones until you see the zone into which you want to import rights and roles.
3. Select the **Authorization** node, right-click, then click **Import Roles and Rights**.

4. Click **Browse** to navigate to the file that contains the authorization information you want to import, then click **Next**.
5. Select the information you want to import, then click **Next**.
6. Review the information to be imported, then click **Finish**.

Copying Rights and Role Definitions into a New Zone

Exporting and importing information from one zone to another is the best solution if you want to include most or all information about rights and roles in a new zone. If you want to limit the information copied from one zone to another, you can copy and paste or drag and drop the information instead. With copy and paste, you can select specific right definitions, role definitions, or role assignments that you want to include in a new zone.

To copy role assignments from one zone to another, however, you should verify that the role definition associated with the role assignment exists in the new zone or is included in the information you are copying to the new zone.

To copy rights, role definitions, or role assignments:

1. Open the Access Manager.
2. Expand **Zones** and the parent zone or child zones until you see the zone that has the rights, role definitions, or role assignments you want to copy.
3. Expand the **Authorization** node.
4. Expand **Window Right Definitions**, **Role Definitions**, or **Role Assignments** until you see the specific right, role, or role assignment you want to copy.
5. Select the specific right, role definition, or role assignment to copy, right-click, then click **Copy**.
6. Open a different zone and expand **Authorization > Windows Right Definitions**, **Role Definitions,** or **Role Assignments**, right-click, then click **Paste**.

Alternatively, you can select a specific right, role definition, or role assignment and drag it to the appropriate node in a new zone.

Viewing Rights and Roles

You can view the status and effective rights for any user in a zone, whether they have been assigned a role or not. You can view detailed information about the rights and role assignments for users by selecting **Show Effective Windows User Rights** in the Access Manager console.

Displaying Rights for an Individual User in the Console

To view role assignments and Windows access rights for a user in the Access Manager console:

1. Open Access Manager.
2. Expand **Zones** and the parent zone or child zones until you see the zone that has the user of interest.
3. Right-click, then click **Show Effective Windows User Rights**.
4. Select a user to see information for the user in the selected zone or click **Browse** to select a specific computer in the zone if you only want to view user rights for a particular computer in the selected zone.
5. Click a tab to see the user's role assignments, desktop rights, application rights, or network access rights.
    - **Role Assignments** lists the user's role assignments, including where the assignment was made. For example, the Object Assigned column indicates whether the assignment for a user is explicit (*user*@*domain*), from a group (*group*@*domain*), or inherited from another setting (All AD Accounts). The Start Time and End Time are only displayed for roles that have time constraints.
    - **Windows Desktops** lists the user's desktop rights granted by the roles to which the user is assigned. The tab identifies the account that can be used to open a new desktop or run an application, the zone where the desktop right is defined, and the role definition that includes the right.
    - **Windows Applications** lists the user's application rights granted by the roles to which the user is assigned. The tab identifies the specific application and the account that can be used to run the application, the zone where the application right is defined, and the role definition that includes the right.
    - **Network Access** lists the user's network access rights granted by the roles to which the user is assigned. The tab identifies the account that can be used to connect to services on a remote computer, the zone where the network access right is defined, and the role definition that includes the right.
6. Click **Close** when you are finished reviewing user rights in a zone or on particular computers.

Scenario: Using a Network Access Role to Edit Group Policies

The steps in this section illustrate a specific scenario of how to configure and use a desktop right and a network access right that allows the user Josh.Adams to log on with his normal Active Directory credentials, open an application that enables him to edit group policies, then connect to a domain controller with administrative privileges so that he can edit a Group Policy Object.

1. Install the Agent for Windows on the domain controller.
2. Install the Agent for Windows on a Windows computer that hosts the Group Policy Management console that the Josh.Adams uses to access the domain controller remotely.
3. Assign Josh.Adams the predefined Windows Login role and the custom role definition gpedit that includes a desktop right and a network access right.
4. Josh Adams logs on to his Windows computer using his Active Directory user name and password.

   To use a role with network access rights, you cannot log on using a local user account. You must use a domain user account authenticated using Active Directory.
5. On his local computer, Josh right-clicks the Delinea icon in the system tray section of the task bar, then selects New Desktop.
6. In his list of available roles, Josh selects his gpedit role, then clicks OK.
7. Josh opens the Group Policy Management console on his local computer, connects to the domain controller in the console, then selects the default domain policy Group Policy Object.
8. Josh right-clicks the default domain policy, then selects Edit to modify the group policy.
9. When he is done working with the group policies, he switches back to his default desktop.

Scenario: Using Multiple Roles for Network Resources

For the local computer, users can select only one role at a time for their desktop or running an application. However, users can select more than one role to access network resources. By selecting multiple roles on the client, users can run applications that connect to multiple remote servers to perform administrative tasks.

In this scenario, Maya.Santiago uses a privileged account to open SQL Server Management Studio on her local computer. From this application, she wants to add accounts that require domain administrator privileges on a remote domain controller and modify database settings on a remote SQL Server instance. To do her work, she needs elevated privileges to run SQL Server Management Studio on her local computer and network access rights to contact the domain controller and the database server.

As the administrator, you have prepared the environment:

- You have put computers in appropriate zones and configured appropriate rights.
- You have configured a role definition, SideBet-DC-Admin, that grants network access to the domain controller using elevated privileges.
- You have also configured a role definition, SQL-DB-Default, that grants network access to SQL Server instances using elevated privileges.
- You have assigned Maya.Santiago to the roles.

To use an application that connects to multiple remote servers:

1. Install the Agent for Windows on the domain controller, the computer that hosts the SQL Server instance, and the computer Maya.Santiago uses to manage the SQL Server instance.
2. Assign Maya.Santiago the custom roles definition SideBet-DC-Admin that includes a desktop right and a network access right.
3. Maya.Santiago logs on to her Windows computer using her Active Directory user name and password.
4. On her local computer, Maya right-clicks SQL Server Management Studio, selects **Run with Privilege**.
5. Maya clicks **Advanced View** to see the list of available roles and selects SideBetDCAdmin as the local role that enables her to run local applications with administrator privileges.
6. Maya then clicks the **Select one or more network roles** option and selects the SideBetDC-Admin role for remote access to the domain controller and the SQLDBDefault role for remote access to the database server, then clicks **OK**.

   After she clicks OK, SQL Server Management Studio starts and she connects to the remote SQL Server instance using Windows authentication. The change to a role with privileges is recorded in the local Windows Application event log.
7. Maya uses SQL Server Management Studio to add and modify information on the domain controller and the SQL Server database.
8. When she is done working, she closes the application and returns to her default desktop and her login account privileges.

Defining Rights for Windows Applications that Encrypt Passwords

Microsoft provides a data protection application-programming interface (DPAPI) to enable applications to secure sensitive information, such as passwords, using encryption. The Data Protection API is the most common way to secure personal information on Windows computers because the information that is encrypted for one user cannot be decrypted by another user. Many applications and system services, including Microsoft Encrypting File System (EFS), Microsoft Internet Explorer, and Google Chrome for example, use the Data Protection API to encrypt passwords.

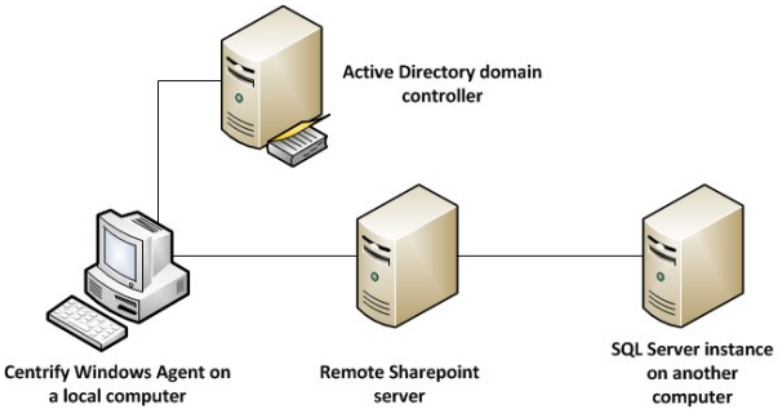
To use a desktop or application right with an application that uses the Data Protection API, you should select the **Self with added group privileges** option for the Run-as account. If you select this option when defining a right, you can install the Agent for Windows on the computer where the application using the Data Protection API is installed to allow users to run the application with administrative privileges.

If you want to use a specific user account for an application that uses the Data Protection API, you must install the Agent for Windows on both the domain controller and the computer where the application using DPAPI is installed. You must also make sure the domain controller is in a zone where users who are going to use the application are granted network access rights. In this scenario, the domain controller must be able to confirm the identity of the specific user account to allow protected information to be decrypted.

For example, assume you define an application right for running Access Manager using the Windows AM-Owner account and assign the user Steve to a role that has this application right. When Steve logs on to the computer where Access Manager is installed and opens the application using the role he is assigned, the Agent for Windows on the domain controller identifies him as the user AM-Owner and provides Jess with the master key for encryption and decryption, enabling him to use Access Manager to add computers, deploy agents, and perform other tasks.

Enabling Access Across Multi-tiered Application Layers

The traditional client/server scenario involves using a Windows client computer to connect to a Windows server to perform some operation. However, it is increasingly common that privileged access must cross multiple application layers. For example, you might have users who log on with their normal credentials who perform administrative tasks on a remote Sharepoint server and those tasks further require access to a SQL Server instance on yet another computer.

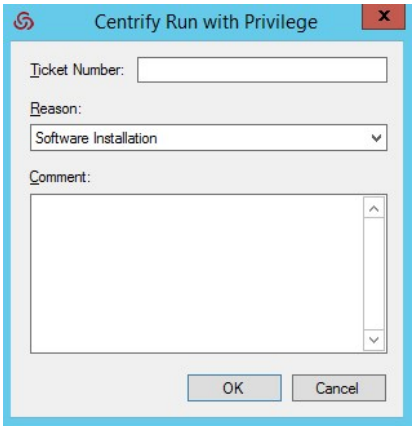


One way to ensure access across multiple applications tiers is to have all of the remote computers involved be in the same zone. At a minimum, the client computer and the computer in the first tier must have the Agent for Windows installed. If the client computer and the computer in the first tier are in different zones, which is the most common scenario, you should place computers in any additional tiers in the same zone as the computer in the first tier.

Requiring Users to Justify Privilege Elevation

You can assign some group policies that force your users to provide a reason when they choose to run an application with privilege. There are two group policies that you can use:

You can use just one of these policies or both. With either of these policies, when a user goes to run an application with privilege, they're prompted with an additional dialog box where they can enter a ticket number, a reason category, and any comments.



The above dialog box prompts users to enter the following information:

- **Ticket number**: If you have enabled the privilege elevation validator policy and subsequent script, you can validate the ticket number that a user enters against a ticketing system such as ServiceNow. If you haven't enabled the privilege elevation validator policy, users can enter any text string here.
- **Reason**: The user selects the reason category that best fits their situation. Their choices are:
  - Software Installation
  - Remote System Administration
  - Local System Administration
  - Windows Feature Management
  - System Networking Change
  - Maintenance (Shutdown, Reboot, Power Off)
  - PowerShell or Other CLI
  - Operation (Services, Zone Operations, etc.)
  - Other
- **Comment**: The user enters any comments about their need to run with privilege. You can view these comments in the audit trail event.

For more details about these policies, see the Group Policy Guide and the group policies' explain text.

Working with Computer Roles

A computer role associates a group of computers in a zone with a set of role assignments to users or groups. For example, you might have a set of computers dedicated to a specific function, such as hosting Oracle databases or payroll processing application. Users who are database administrators for those computers require different privileges than users who update payroll records on those computers.

Using a computer role, you can associate the group of computers that host an Oracle database with a specific role assignment, for example, users who are assigned the oracle dba role. The oracle-dba role definition might include desktop and network access rights because the users assigned to the oracle-dba role require administrative privileges.

You could also create a second computer role that associates the group of computers that host the payroll processing application with a group of users who are allowed to log on and update payroll records without granting any other administrative privileges. For example, if some of the computers that host an Oracle database are used for payroll processing, you can define another computer role—payroll-west—that associates just those computers with the role assignment payroll_mgmt. The payroll_mgmt role definition might have the console login right and an application right specifically for the payroll application. When users are assigned the payroll_mgmt role, they can log on locally and run the payroll application with elevated privileges only on the group of computers defined in the computer role payroll-west.

To use computer roles, you must do the following:

- Decide on the attribute the computers in a particular group share. For example, you can use a computer role to identify computers in the web farm, that host specific applications, or serve a specific department.
- Identify the sets of users that share common access rights and create Active Directory groups for them. For example, if you are creating a computer role for Oracle database servers, you might have different access rights for application users, database administrators, and backup operators.
- Identify the role definitions each set of users should be assigned. For example, application users role might use the default Windows Login role, while administrators might require a custom role definition with desktop and network access rights, and backup operators might require a custom role

definition with an application right.

Using Computer Roles to Simplify the Management of Access Rights

Deciding how best to use computer roles requires some planning and configuration that may not be part of your initial deployment plan. To make effective use of computer roles, you also prepare appropriate role definitions for different sets of users. However, computer roles provide a powerful and flexible option for managing access to computers using your existing processes and procedures for managing Active Directory group membership.

After you create a computer role, it is easy to manage even as your organization changes and grows. For example, if another Oracle database server comes online, you add it to the computer group you created for Oracle database servers in Active Directory. If other DBAs join your organization, you add them to the Active Directory group you created for Oracle administrators. The computer role links the computer group to the role assignment and no additional updates are needed to accommodate these kinds of organizational changes. If you need to modify the access rights, you can change the role definition and have the changes apply to all members of the group.

Create an Active Directory Group for a Set of Computers

Computer roles create links between objects in Active Directory and access rights defined in Access Manager. After you have identified a group of computers that share a common attribute, you should create an Active Directory group for those computers if one does not already exist.

You can also create the computer group and add its members directly from Access Manager when you create the computer role. If you are not preparing the Active Directory group before creating the computer role, you can skip this section and go directly to Create a new computer role.

To create an Active Directory group for computers in a computer role:

1. Open Active Directory Users and Computers to create a new Active Directory group.
2. For example, create a new Active Directory group for Oracle Database Servers.
3. Select the new computer group, right-click, then click **Properties**.
4. Click the **Members** tab, then click **Add**.
5. Click **Object Types**, select **Computers**, then click **OK**.
6. Search for and select the computers that you have identified as Oracle database servers as members of the new group, then click **OK**.
7. Click **OK** to save the group.

Create an Active Directory Group for Each Set of Access Rights

In addition to the Active Directory group for the computers in a computer role, you should have an Active Directory group for each set of users that should have different access rights. By mapping Active Directory groups to role definitions, you can manage group membership and access rights at the same time using your current procedures.

To create an Active Directory group for each set of users linked to a computer role:

1. Open Active Directory Users and Computers to create a new Active Directory group for each set of users to link to the computer role.

   For example, create separate Active Directory groups for application users, database administrators, and backup operators using a naming convention similar to *ComputerAttribute*_Role_*UserSet*. For example, create the following Active Directory groups:

   - OracleServers_Role_AppUsers
   - OracleServers_Role_DBAs
   - OracleServers_Role_Backup

2. Select each new group, right-click, then click **Properties**.
3. Click the **Members** tab, then click **Add**.
4. Search for and select the users that you have identified as members of the each group, then click **OK**.
5. Click **OK** to save the group membership.

Create a Role Definition for Each Set of Users with Different Access Rights

Before you create a new role definition, identify the specific rights associated with each role and define those rights if they do not already exist. For this sample scenario, you might create role definitions similar to the following:

- Oracle_AppUsers with Windows Login access and an application right for a specific database application.
- Oracle_DBAs with Windows Login access and desktop and network access rights on computers in a specific zone.
- Oracle_Backup with console login allowed right and an application right that allow members of the group to run backup utilities with the privileges of the built-in Backup Operators group.

Create a New Computer Role

After you have prepared the appropriate Active Directory groups and role definitions for different sets of users, you can create one or more computer roles.

To create a new computer role:

1. Open Access Manager.
2. Expand **Zones** and the parent zone or child zones until you see the zone that has the computer for which you want to define a computer role.
3. Expand the **Authorization** node.
4. Select **Computer Roles**, right-click and click **Create Computer Role**.
5. Type a name and description for the computer role.

   For example, type OracleServers, and an optional description, such as Oracle database servers in the San Francisco data center.
6. In **Computers group** list, select **<...>** to search for the Active Directory group of computers you created in Create an Active Directory group for a set of computers.

   Select **<Create group >** if you want to create a new Active Directory group of computers and add members now. If you are creating a new group, click **Browse** to select a container to use, type a group name, and select the scope of the group, then click **OK**.
7. Click **OK** to save the computer role.
8. If you selected an existing computer group, expand **Computer Roles > Members** to see the computers that are members of this computer role.

   If you created a new computer group at Step 6, select the new computer role, right-click **Members**, then select **Add Computer** to search for and select one or more computers to add to the group.

Add Role Assignments to the Computer Role

If you have created the appropriate Active Directory groups and role definitions that you want to assign, you can now assign the roles to set of users as required.

To add role assignments to users in Active Directory groups:

1. Expand the computer role you just created, for example, expand **OracleServers**.
2. Select **Role Assignments**, right-click, then click **Assign Role**.
3. Select the role definition from the list of roles, then click **OK**.

   For example, select the Oracle_DBAs role definition. By default, the role is set to start immediately and never expire. You can set a **Start time**, **End time**, or both start and end times for the role assignment. For example, if the role applies to a contractor who will be hired for a specific amount of time and you want to automatically disable the role after they finish the job and leave the organization, you can specify the start and end times when you assign the role.
4. Select whether the role assignment applies to all Active Directory accounts, all local accounts, or specific Active Directory and local accounts, then click **OK** to complete the role assignment.

   For example, to assign the Oracle_DBAs role to the Active Directory OracleServers_Role_DBAs security group, click **Add AD Account**. You can then select **Group** to search for the group, select it from the results, then click **OK**.
5. Repeat Step 1 through Step 4 for each group that you want to add to this computer role. For example, repeat the steps to assign the Oracle_AppUsers role to the OracleServers_Role_AppUsers security group and the Oracle_Backup role to the OracleServers_Role_Backup security group.
6. Select the **Role Assignments** node to see all of the role assignments you have defined for groups associated with the computer role.

7. Select the **Members** node to see the computers or groups of computers to which the role assignments apply.

Assigning Roles on Multiple Computers at Once

To simplify the process of assigning Active Directory users or groups to a role, you can perform a bulk role assignment. With a bulk role assignment, you can assign a role to multiple Active Directory users and groups on multiple computers at the same time. For example, if you have two groups of SQL Server administrators and three computers where the members of those groups need access to their SQLServerAdmin role, you can select those two groups and those three computers to be assigned the SQLServerAdmin role in the same process. You can also specify optional start and end times for the role assignment and have those settings apply for all of the users, groups, and computers you have selected for bulk assignment.

To assign users and groups to a role in a zone:

1. Open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone where you want to make role assignments.
3. Right-click, then select **Assign Roles to Computers**.
4. Type the user and group names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for user and group names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria.
5. Type the computer names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for the computer names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria.
6. Select a role for the list of roles available, then click **OK**.
7. Review the role assignment start and end time and the user and group accounts that are being assigned the role, then click **OK**.

   You can make changes to the start and end times if you want those changes applied for all of the users, groups, and computers that are part of this bulk role assignment.

After you click **OK**, the selected users and groups are then automatically assigned the selected role on the selected computers.

Using the Authorization Center Directly on Managed Computers

The Authorization Center is available on managed computers where you have deployed the Agent for Windows and enabled the privilege elevation service. From the Authorization Center, you can view details about the rights, role definitions, role assignments, and auditing status for any users. Individual users can see details about their own login rights, effective roles, role assignments, role definitions, and auditing status. Administrators can select any user of interest to view the details for that user.

To use the Authorization Center on a local computer:

1. Log on to a computer where the Agent for Windows and privilege elevation features are deployed.
2. Click the arrow next to the notifications area in the taskbar.
3. Right-click the Delinea icon, then select **Open Authorization Center**.
4. Click a tab to see details about the current user's roles.
   - **Effective Login Rights** displays the current user's local, remote, and PowerShell login rights and whether auditing is requested, required, or not applicable.
   - **Effective Roles** lists the roles that have been assigned to the current user and the status of each role names to which the user is assigned. You can right-click a role, then select Role Properties to view additional details, such as any time constraints defined for the role and the specific rights granted by the role.
   - **Role Assignments** lists details about the user's role assignments, including where the assignment was made. For example, the Object Assigned column indicates whether the assignment for a user is explicit, from a group, or inherited from another setting, for example, from the selection of All Active Directory Accounts. You can right-click a role, then select Assignment Properties or Role Properties to view additional details, such as any time constraints defined for the role and the specific rights granted by the role.

- **Role Definitions** lists detailed information about the selected user's login rights and the audit requirements that have been defined for the roles the user has been assigned. You can right-click a role definition, then select Properties to view additional details.
- **Auditing** lists the desktops used and auditing status for each desktop started in a session.

5. Click **Browse** to view information for another user.
6. Type all or part of the user name, then click **OK**.

   If more than one user name is found, select the appropriate user from the results, then click **OK**.
7. Click **Close** when you are finished viewing detailed authorization information for the selected user.

Working with the Authorization Cache on Managed Computers

Authorization information—such as your rights, role definitions, and assignments—is cached locally on each computer where you have deployed the Agent for Windows. The cache saves access privilege information to improve performance and also to persist elevated privilege capabilities for users and groups when the computer is not connected to Active Directory.

The following sections describe:

- Which Server Suite capabilities are and are not persisted by the cache when a computer is disconnected from a domain controller.
- Where the cache resides.
- How and when to perform cache operations such as refreshing, flushing, and dumping.

Persisted and Non-persisted Capabilities

The Server Suite cache persists several role-based capabilities when a computer is not connected to Active Directory. A computer is considered to be *not connected* when the Windows agent is unable to reach one or more of the following entities:

- The domain to which the computer is joined.
- The domain of any zone in the *zone hierarchy*. The zone hierarchy is the domain of the zone that the machine is joined to, or any parent zones of that joined zone.
- An Active Directory global catalog (GC) associated with any of these domains.

If the Windows agent can reach all of these entities, it is considered to be *connected.*

Persisted Capabilities

These capabilities are supported when a computer is not connected:

- Users can log in based on role.
- Users can run applications based on role.
- Users can create desktops based on role.
- Computers can be removed from zones.
- Delinea software can be installed (but the computer cannot be joined to a zone).
- Delinea software can be upgraded, but this practice is not recommended because there will be no authorization data in the cache after the upgrade.

Non-persisted capabilities

These limitations exist when a computer is not connected:

- You cannot join a zone or change a computer's zone.
- The use of Network rights is not supported.

Cache Location

The cache resides in <!---TODO update path ---> *SYSTEMDRIVE*\ProgramData\Centrify\DirectAuthorize\Cache.

Performing Cache Operations

You must have administrator privileges to perform the cache operations described here. Available cache operations include:

- Refreshing the cache (perform this operation from the user interface or the command line)
- Flushing the cache (performed from the command line)
- Dumping the cache (performed from the command line)

Refreshing the Cache

As administrator, you can refresh the cache from the user interface or from the command line. Refreshing the cache updates the cache with fresh information from Active Directory, ensuring that the agent has the most up-to-date information about users' current rights and roles.

Refreshing the cache is useful if you change authorization information with the management console, and you want to see the updated information on the Windows agent right away.

> **Note**: In domains containing multiple domain controllers, you might not see the updated information even after you refresh the cache. In cases such as this, wait for Active Directory replication (typically a few minutes), and then refresh the cache again. Alternatively, wait another 10 minutes and the agent will refresh the data on its own.

You can refresh and flush the cache only on computers that are connected to a domain controller.

To refresh the cache from the user interface:

1. Open the agent configuration panel by clicking **Agent Configuration** in the list of applications on the Windows Start menu.
2. Click **Privilege Elevation Service**.
3. Click **Settings**.
4. Click the **Troubleshooting** tab.
5. Click **Refresh**, then click **OK** to acknowledge the successful operation.

> **Note**: Alternatively, you can execute the dzrefresh command line utility to refresh the cache as described in the next section.

To refresh the cache from the command line:

Execute the dzrefresh command line utility to refresh the cache. Executing dzrefresh performs the same operation as clicking the **Refresh** button in the agent configuration panel **Troubleshooting** tab.

The syntax for running the dzrefresh utility is:

dzrefresh

Flushing the Cache

Execute the dzflush command line utility to flush (clear) the cache. Flushing the cache removes all cache data and reloads it from Active Directory. You should flush the cache only when directed to do so by Support. Under most circumstances, you should refresh the cache rather than flush the cache.

The syntax for running the dzflush utility is:

dzflush

Dumping the Cache

Execute the dzdump command line utility to dump the cache to standard output or to a redirect file that you specify on the command line. You can also use the options shown here to display only specific types of cache data, such as zone hierarchy, role definitions, right definitions, and other data.

You should execute the dzdump utility only when directed to do so by Support.

The syntax for running the dzdump utility is:

dzdump [/d [*directory-path*]] [/w=*screen-width*] [/s] [/n] [/g] [/l] [/a] [/r] [/i] [/t] [/z] [/u] [/h]

If you execute dzdump with no options, all dzagent in-memory cache is dumped.

Setting valid options

You can use the following options with dzdump:

/d	Dump cache files from the default location.
/d=*directory-path*	Dump cache files from the specified location.
/w=*screen-width*	Use the specified width rather than the default of 80 for word-wrap. Set /w=0 to disable word-wrap.
/s	Display SID mappings.
/n	Display name mappings.
/g	Display assignee mappings.
/l	Display assignments in the joined zone hierarchy.
/a	Display assignments for SIDs.
/r	Display role definitions.
/i	Display right definitions.
/t	Display access token information.
/z	Display zone hierarchy.
/u	Display recent user log-ins.
/h	Display help information.

Configuring PowerShell Remote Access

You can run PowerShell commands on remote computers and have the agent handle the authentication and privilege elevation for you. In order to run remote PowerShell commands, the following requirements apply:

- The target computer needs to have the Agent for Windows installed with the Privilege Elevation Service enabled.
- Assign the user to a role with the "PowerShell remote access is allowed" system right granted.

If you're using the Audit & Monitoring Service, when a user attempts to run PowerShell remotely on a computer, the system triggers an audit trail event. Audit & Monitoring Service is an optional service.

To assign PowerShell remote access to a user:

1. In the **Access Manager** console, open the zone that the Windows system to be managed belongs to (Access Manager is not necessary installed on the machine with the Windows agent).
2. Under **Role Definitions**, right-click a role that you'd like to assign PowerShell remote access permission to and select **Properties**.
3. Under **System Rights** > **Windows rights**, select **PowerShell remote access is allowed**.
4. Right-click **Role >Assignment** and select **Assign Role**.
5. Select the role as defined above and assign the Windows account to it.

What Gets Audited for Remote PowerShell Commands and Scripts

For cases where someone runs individual PowerShell cmdlets, the audit trail event captures the following details:

- Specific cmdlets that were run
- Arguments
- Return codes

- User who ran the cmdlets
- The timestamp when the user ran the cmdlets

For cases where someone runs a PowerShell script, the audit trail event captures the name of the script as well, and if the script was run remotely the audit trail event captures the contents of the script. If the script is very long, the audit trail will truncate it and add an ellipsis (...).

> **Note**: If the user runs a PowerShell script on the target system from that same system, the audit trail event does NOT capture the contents of the script. This is due to a limitation in Windows Remote Management. Basically, the thing to remember is that if you send over script text to a remote system, the audit trail captures the script text; if you send over just a script filename, that's what the audit trail captures.

Examples of Remote PowerShell Commands

For example, if a user runs individual PowerShell commands on a remote system, they would start the session with a command similar to the following:

Enter-PSSession -ComputerName **targetcomputername**

The audit trail event captures details about any commands that the user enters during the above PowerShell session.

As another example, if a user runs a script without first creating the remote session and runs the script against a remote, target system from another system, the user might run a command similar to the following:

Invoke-Command -ComputerName **targetcomputername** -FilePath {c:**script**.ps1}

In this second example, you'll know that the user ran a script because there'll be a isscript=true parameter in the audit trail.

As a final example, if a user runs a script without first creating the remote session and runs the script from the target system, the user might run a command similar to the following:

Invoke-Command -ComputerName **targetcomputername** -Command{c:**script**.ps1}

Hiding the Remote PowerShell Script Text

There may be situations where your users have scripts to run on remote systems but you don't want or need the script text to appear in the audit log. To hide the script text from the audit log, change the following registry to 1 (the default value is 0): <!---TODO update path --->

SOFTWARE\Policies\Centrify\DirectAuthorize\Agent\HideRemotePsScript (REG_DWORD)

You can set the HideRemotePsScript option by group policy.

Authentication Service Enforcement

Any time you open the Access Manager console, a background process determines the availability of licenses.

As you increase the number of licenses in use, license enforcement is progressive. If the number of computers is less than 90% of the number of licenses you have purchased, there's no affect on any auditing features. If the number of computers is more than 90% of the licenses purchased, enforcement depends on the number of licenses in use:

- 90-100% of the licensing limit displays a warning message that you are close to over deployment, but you can continue to use all authentication and privilege elevation features.
- 100-120% of the licensing limit displays a warning message that you must acknowledge by clicking **OK** when you open the console, after which you can resume using the console.
- Over 120% of the licensing limit displays a warning message for 60 seconds when you open the console. If you see the 60 second warning message, use the License dialog box to add license keys to continue using features.

You can contact Delinea to purchase additional licenses or remove some computers to bring the number of licenses used into compliance.

Configuring MFA with RADIUS for Privilege Elevation Service for Windows Checklist

This document provides a configuration checklist for 3rd party multi-factor authentication providers such as Duo, Okta, SecurID (or any other vendor that provides a RADIUS service) to provide identity validation with the Privilege Elevation Service in the Microsoft Windows platform.

If you have an identity service provider (such as Duo, Okta, SecureID, and so forth) that you use for MFA logins, you can integrate authentication and privilege

elevation with your identity provider and the RADIUS protocol to require MFA for privilege elevation tasks, such as Run with Privilege and New Desktop.

Make sure that you work with your RADIUS expert along with your network and directory services lead administrators during the design and configuration tasks.

The checklist below includes links to documented procedures.

> **Note**: If you use Privileged Access Service, although you can enable MFA with RADIUS, the recommended practice is that you use the native integration.

	RADIUS requirements	
1	Gather the following settings for your RADIUS service: IP address or fully qualified domain name Port Timeout settings Pre-shared secret	
2	Verify that you can generate a RADIUS one-time password successfully.	
3	Verify that identity authentication is working correctly with your RADIUS system.	
4	Have access to someone who is knowledgeable about your RADIUS system and can answer questions or help troubleshoot issues, if needed.	
	Windows and Active Directory requirements for RADIUS configuration	
5	A Windows computer to use as a RADIUS client for initial testing, including: Client name Client IP address	
6	Make sure that client systems can reach the RADIUS server over the network (check your firewall settings). You may need help also from your network team if your RADIUS cluster has a load-balancer in the front end.	
7	You have administrative access to the designated Windows computer so that	

	you can install software and do configurations.	
8	You have Active Directory account access so that you can modify group policies that apply to the target computer.	
9	You have access to the Group Policy Management Console.	
10	Your Active Directory expert must decide how the group policy layout and scope will be designed so that the group policies are applied to the clients based on their RADIUS service availability.	
	Authentication and Privilege Elevation Services Requirements for RADIUS configuration	
11	Access Manager console is installed on the client computer.	For details, see Running the setup program on a Windows computer.
12	The Agent for Windows is installed on the client system, you've configured the system to work with Privilege Elevation Service, including joining the computer to a zone.	For details, see Installing the Agent for Windows.
13	You have administrative access to Access Manager so that you can manage roles and rights.	
14	The group policy templates from release 19.6 or later are installed. For RADIUS configuration, you need at least the Delinea Windows settings group policies.	For details, see Installing group policy extensions separately from Access Manager.
15	If you want to capture the RADIUS events in your SIEM system, make sure the Audit trail is configured to go to the local log file.	In GPME, go to computer Configuration > Policies > Audit Trail Settings > Global Settings > Send audit trail to log file (this is not configured by default). For details, see "Send audit trail to log file" in the *Group Policy Guide*.

16	You have a role and user to test with. Make sure the role has rights for privilege elevation, such as New Desktop rights or Run as Role.	Make sure that you can elevate privileges successfully for that user and role before you try to configure RADIUS authentication.
	Configure a system to use RADIUS for privilege elevation (using group policies)	
17	Enable and configure the RADIUS group policies.	Configure the following group policies: Windows › MFA Settings › Specify the authentication source for privilege elevation : set this policy to RADIUS Authentication. Windows › MFA Settings › Remote Authentication Dial-In User Service (RADIUS) Settings › Enable Remote Authentication Dial-In User Service (RADIUS): enable this policy. Specify the RADIUS connection timeout: Configure to match your RADIUS timeout setting. Specify the RADIUS server IP address: enter your RADIUS IP address. Specify the RADIUS server port number: enter your RADIUS port number (the default is 1812). For details, see "Remote Authentication Dial-In User Service (RADIUS) Service Settings" in the *Group Policy Guide*. After you update the policies, do a group policy update on the Windows client computer.
18	Configure the role to require re-authentication using multi-factor authentication.	For example: Right-click your test role and choose Properties. The Role Properties dialog box opens. Click the Run As tab. Select Re-authenticate current user and then select Require multi-factor authentication. Click OK to apply the changes.
19	Run dzflush to make sure that the agent has the changes from Access Manager.	For details, see Using dzflush.
20	Set the RADIUS shared secret.	The RADIUS secret is unique to each system and will match the secret that the RADIUS server has. You can set the pre-shared secret by either of the following methods on the client computer: Run the Set-CdmRadiusSecret cmdlet to set the RADIUS shared client secret. For details, see the DirectAuthorize PowerShell cmdlet help. Use the Agent Configuration settings dialog box to configure the RADIUS server, including the pre-shared secret. For details, see Configuring agent settings for the Identity Platform.
	TEST AND VERIFY	
21	Verify that a user can elevate privileges by entering the RADIUS one-time password.	For example, if your role has New Desktop rights: Right-click the System Tray and select New Desktop. In the dialog box that appears, select your test role and click OK. If the RADIUS authentication has been configured successfully, you are prompted to enter a password for RADIUS authentication. Enter the password and click Next to continue. You can also view the audit trails for the successful authentication in the system's event log.
22	Verify that a user cannot elevate privileges after entering an incorrect RADIUS one-time password.	

Adding Remote Users Automatically

If desired, you can configure your deployment so that members of the Windows Login group or the Windows Remote Login group are also automatically added to the Remote Desktop Users group and the ConsoleLogonUser group.

To make this change, add the following registry entry on each computer where you have installed the Agent for Windows: <!---TODO update path --->

HKEY_LOCAL_MACHINE\SOFTWARE\Centrify\DirectAuthorize\Agent\EnableAddUsersToRdpAndConsoleLogOn = 1

If you later uninstall the agent, the uninstall process removes the affected user accounts from the Remote Desktop Users group and the ConsoleLogonUser group. Only the user accounts that the agent added to those groups are affected.

Enabling Users to Run Applications with Alternate Accounts

Alternate accounts are typically a privileged or administrator account in Active Directory that's associated with an owner account. You can log in to the alternate account using your main account.

For example, system administrators typically have several accounts, a user account for general log-ins and an administrative account to access specific systems and services.

Here are the things you need to do in order to enable the ability to run with alternate accounts:

1. Set up alternate accounts for users in Privileged Access Service
2. Install a cloud connector in your domain and the Web Server (IWA) service is enabled.
3. Enable the policy entitled "Enable run with alternate account."
4. (Optional but recommended) Configure the following policies to set up a grace period after which time users running applications with alternate accounts must re-authenticate:
    - "Require re-authentication to run application with alternate account"
    - "Configure Windows authentication grace period for run with alternate account"
5. Install the Agent for Windows and enable the Identity Platform service on each computer where you want users to be able to run with alternate accounts.

If you don't enable the run with alternate account feature, your users can still run applications with these alternate accounts by logging in to Privileged Access Service and checking out the password.

You can manage your local Windows users and groups, if desired. This way, you can centrally manage the accounts.

Overall, to manage local users and groups on Windows systems, you'll need to

- Install the Agent for Windows on each Windows system where you want to manage local accounts.
- Enable local account management on those Windows systems in the Privilege Elevation settings for the agent. For details, see Enabling Windows local account management.
- In Access Manager, you can then add, edit, or remove local users and groups. For details, Adding local Windows accounts and Removing local Windows accounts
- Manage the passwords for local Windows accounts. For details, see Creating and managing local Windows user passwords.
- Use group policies to manage local Windows accounts. .

Adding Local Windows Accounts

Before you enable local account management on your Windows computers, add the local users and groups in Access Manager.

Note: {/b}If you first enable local account management with the enforce option and if you have any existing local accounts on that system but not defined in a zone, then the service will remove those local users during the next synchronization. Built-in local Windows accounts are not removed.

To add a local Windows user:

1. In Access Manager, navigate to either a zone or a Windows computer and go to **Windows Data**
2. Right -click **Local Users** and select **Add User to Zone** or **Add User**, depending on where you're adding the user.
3. Enter the user name and click **OK**.
4. Specify the attributes for the local Windows user:
    - **Full name**: The first and last name of the new local Windows user.
    - **Description**: A description of the user.
    - **State**: Specify one of the following:
        - **Enable**: Set the state to Enable for a local Windows account that is in use.
        - **Disable**: Set the state to Disable for a local Windows account that is not in use.
        - **Remove**: If you've chosen not to enforce local account management, mark the user as Remove and the service will remove the user at the next synchronization interval.

          Note: {/b}The service will not remove any built-in local Windows accounts, even if you mark it as Remove in Access Manager.
    - **Password options**: If desired, select any of the following:
        - **User must change password at next logon**: The service will force the local Windows user to change the account password the next time that the user logs in to the computer. Note that this option applies only to new accounts.
        - **User cannot change password**: The user won't be able to change the password.
        - **Password never expires**: The user's password will never expire.
5. Click **OK** to save your changes.

   The new user will be available on the affected systems after the next local account synchronization.

To add a local Windows group:

1. In Access Manager, navigate to either a zone or a Windows computer and go to **Windows Data**
    1. Right -click **Local Groups** and select **Add Group to Zone** or **Add Group**, depending on where you're adding the group.
    2. Enter the group name and click **OK**.

2. Specify the attributes for the local Windows group:
    - **Description:** Enter a description of your choice.
    - **Members:** Click **Add** to launch the Add Members dialog. In a comma-separated list, type the names of the users who will be in the group.

      Note that Access Manager does not check the validity of the user names that you provide. You should ensure that all of the names that you provide are local Windows user names that currently exist.
    - **State:** Specify either **Enable** or **Remove**.
        - **Enable**: Set the state to **Enable** for a local Windows account that is in use.
        - **Remove**: If you've chosen not to enforce local account management, mark the group as **Remove** and the service will remove the group at the next synchronization interval.
3. Click **OK** to save your changes.

   The new group will be available on the affected systems after the next local account synchronization.

Enabling Windows Local Account Management

You can have Delinea manage your local Windows user and group accounts; to do so, you need to enable and configure a few settings. Install the agent and enable local account management on each Windows system where you want to manage local accounts.

Be aware that if you enable local account management, the service does not delete any built-in Windows users or groups, even if you mark one of those accounts for remove.

Note: {/b}Windows local account management is not supported on domain controllers.

To configure local account management for Windows:

1. From the Privilege Elevation Service Settings dialog box Local Account Management tab, click **Configure**.

   The Local Account Management Configuration dialog box opens.
2. Select the **Enable local account management** option.
3. Select **Yes** to enforce local account management or **No** to not enforce local account management.

   Enforcing local account management means that after you remove a local Windows user or group from Access Manager, the service will remove the local user or group from the computer after the next synchronization.

   If you choose not to enforce local account management, in order to remove a user you mark it as removed rather than explicitly removing the account from Access Manager.
4. Specify a script that will run when the service synchronizes local account information with Access Manager and the affected computers. The script can set the passwords for the local accounts and also display a list of enabled, disabled, or removed users.

   For details, see Creating and managing local Windows user passwords.

   There is a sample script provided that you can use as a starting point:

   C:\Program Files\Centrify\Agent for Windows\SampleNotification.ps1

   The script will run after each synchronization of local accounts when the any of the following have occurred:
    - New local users are added
    - Local users are enabled
    - Local users are disabled
    - Local users are removed
5. Specify a synchronization interval.

   This interval controls how often the service synchronizes local account information between Access Manager and the affected computers. The default is 60 minutes.

6. Click **OK** to save your changes and close the dialog box.

Creating and Managing Local Windows User Passwords

After you create local Windows users, you still need to assign a password to each user. Instead of manually setting the passwords in Local Users and Groups, you'll set up the initial passwords for your local user accounts by way of a PowerShell script.

There is a sample script provided that you can use as a starting point:

C:\Program Files\Centrify\Agent for Windows\SampleNotification.ps1

In general, the script should both set passwords and notify you of changes in local accounts. The script will run after each synchronization of local accounts when the any of the following have occurred:

- New local users are added
- Local users are enabled
- Local users are disabled
- Local users are removed

Typically, the script should perform the following user account tasks:

- Assign a random password to newly provisioned local users.
- Provide the user account information, including the generated passwords, to your password management solution.

After you have the script set up, you can use group policy to automatically run it. .

How you set up the passwords and the script depends on if you're using a password management system or not. Below are the ways you can set up local user passwords.

Use Privileged Access Service to manage local Windows account passwords:

1. Register for Privileged Access Service.
2. Download the Client for Windows software package.
3. On each Windows computer where you will assign passwords to local users, run the cenroll command to register the computer as a managed resource.
4. Create a PowerShell notification script that runs on each of these Windows computers, gives each user a random password, and sends the password to Privileged Access Service.

   In the script, you can set it to run the csetaccount command to send the password to Privileged Access Service.
5. Using one of the following two methods, configure the notification script to run after the agent synchronizes local account information:
   - In the local account management settings for the agent

     Agent settings > Local Account Management tab > Configure > Local Account Management Configuration dialog box
   - In the group policy

     (Settings > Windows Settings > Local Account Management > Notification Command Line)

Use a third-party system to manage local Windows account passwords:

1. Create a PowerShell script that runs on each of these Windows computers and gives each user a random password.
2. Include a section in the script that submits the passwords to the password management product for storage and maintenance.
3. Using one of the following two methods, configure the notification script to run after the agent synchronizes local account information:
   - In the local account management settings for the agent

     Agent settings > Local Account Management tab > Configure > Local Account Management Configuration dialog box
   - In the group policy

(Settings > Windows Settings > Local Account Management > Notification Command Line)

Removing Local Windows Accounts

If you have enabled local account management on a Windows system, there are two different ways to remove users. Your approach depends on if you've configured to enforce local account management or not.

Be aware that if you enable local account management, the service does not delete any built-in Windows users or groups, even if you mark one of those accounts for remove.

To remove a local Windows user or group if local account management is enforced:

- In Access Manager, right-click the user or group and select **Delete**.

  The account is removed from Access Manager immediately. When the service next synchronizes local account information, the service removes the user or group from the affected Windows systems too.

To remove a local Windows user or group if local account management is not enforced:

- In Access Manager, right-click the desired user or group and select **Change Profile State**, then select **Remove**.

  The account is marked as "Remove" and remains visible in Access Manager. When the service next synchronizes local account information, the service removes the user or group from the affected Windows systems too.

This chapter describes how to use the Master Auditor role and group policies to control who is audited and who can search and play back captured user sessions for an installation.

Configuring Selective Auditing

If you are using identity and privilege management features, you can control audit and monitoring service by using Access Manager to configure role definitions with different audit requirements, and then assigning those role definitions to different sets of Active Directory users. For more information about using role definitions to control auditing, see Defining custom roles with specific rights.

If you are using audit and monitoring service without also using identity and privilege management features, you can use group policies to control which Windows users to audit, or to capture activity for all Windows users.

To control audit and monitoring service using group policies:

1. Open the Group Policy Management console.
2. Expand the forest and domains to select the Default Domain Policy object.
3. Right-click, then click **Edit** to open Group Policy Management Editor.
4. Expand Computer Configuration > Policies, then select **DirectAudit Settings**.
5. Select the Audited user list to identify specific users to audit.

   When you enable this group policy, only the users you specify in the policy are audited. If this policy is not configured, all users are audited.
6. Select the Non-audited user list to identify specific users that should not be audited.

   When you enable this group policy, only the users you specify are not audited. If this policy is not configured, all users are audited. If you enable both the Audited user list and the Non-audited user list policies, the users you include in the Non-audited user list take precedence over the Audited user list.

The following table details the effect of configuring and enabling the Audited user list and Non-audited user list group policies, and including or not including Windows users in those lists.

Not configured	Not configured	No users are defined for either policy, so all users accessing audited computers are audited.
Not configured	Enabled	Only the users you specify in the Audited user list policy are audited. If no users are specified when the policy is enabled, no users are audited.
Not configured	Enabled	Only AUL is enabled, but user is not listed in it.
Enabled	Not configured	If no users are specified in the Non-audited user list and the policy is enabled, no users are exempt from auditing. All users are audited.
Enabled	Enabled	If both policies are enabled, the non-audited user takes precedence over the audited list of users. If a user is specified in the audited list, that user is explicitly audited. If a user is specified in the non-audited list, that user is explicitly not audited. If the same user is specified in both lists, the user is not audited because the non-audited user takes precedence. If no users are specified for either policy, all users are audited because the non-audited user takes precedence.

Enabling Audit Notification

If you enable audit notification, users see a message informing them that their actions are being auditing when they log on. fter you enable notification, the

message is always displayed on audited computers if the session activity is being recorded.

To enable audit notification for an installation:

1. In the Audit Manager console, right-click the installation name, then select **Properties**.
2. Click the **Notification** tab.
3. Select **Enable notification**.

   Deselect this option to turn off notification.
4. Click the browse button to locate and select a text file that contains the message you want to display.

   A notification message is required if you select the Enable notification option. The contents of the file you select are displayed below the file location. The maximum text file size is 30 KB.
5. Click the browse button to locate and select an image to appear as a banner across the top of the audit notification.

   Displaying a banner image is optional when you enable notification. The maximum image file size is 15 KB. For the best image display, use an image that is 468 pixels wide by 60 pixels high.

   Note: {/b}Animated GIF files are not supported for use as audit notifications. If you do specify an animated GIF, the image displays as a static image.
6. Click **OK** or **Apply**.

   Users will see the notification message the next time they log in.
7. If you enable notification after you have deployed agents, update the local policy on the audited computers by running the following command:

   gpupdate /FORCE

Managing Audit Roles and Auditors

Audit roles grant access to auditors to search, replay, and delete specific audited sessions using the Audit Analyzer console. Each audit role identifies a set of audited sessions, the list of auditors who have access to those sessions, and what the auditors in a specific role are allowed to do.

You identify a set of sessions by specifying criteria you want to use, for example, all sessions from a particular audited computer, associated with a specific application, or recorded during a specific period of time.

You identify the auditors for a set of sessions by specifying individual Active Directory users or Active Directory groups of auditors. If you use Active Directory groups, you can manage the privileges for all of the members of the group using your existing procedures for managing Active Directory groups. You can also configure the type of access granted to each member of the audit role.

You create and assign users and groups to audit roles using the Audit Manager console. You create the audit roles by rightclicking on the Audit Roles node. You add users and groups to an audit role by right-clicking on the specific role name.

Every installation automatically has a Master Auditor role. The Master Auditor has access to all audit data and permission to read, replay, update the review status, and delete sessions for the entire installation. The Master Auditor can also create roles, assign users, set permissions, and delegate administrative tasks for all of the audit stores in the installation. You cannot rename, delete, or modify permissions for the Master Auditor, but you can assign other users and groups to the Master Auditor role.

Granting Permission to Manage Audit Roles

The Master Auditor can grant the Manage Audit Role permission for an installation to one or more audit team leaders. The Manage Audit Role permission grants full control over all of the audit roles in the installation. An audit team leader can then create new roles, change the permissions specific audit roles grant, add or remove members, and remove roles.

When creating an audit role, an audit team leader defines the following:

- Target session type and optional other criteria.
- A collection of rights on the target sessions: Read, Update Status, Replay, and Delete.

For example, an audit team leader might define the following audit roles to control what different team members can do:

- A role named Windows Session Viewer for first level reviewers with a target of Windows sessions and only the right to Read session information. The members of the First Review group who are assigned to the Windows Session Viewer audit role can read, but not delete, replay or update the status of Windows sessions in the installation.
- A role named Incident Escalation for security managers with a target of Windows sessions from the last 72 hours, and permission to Read, Replay, and Update Status for the targeted session. The members of the Security group who are assigned to the Incident Escalation audit role can read, replay, and update the review status of Windows sessions from the previous 72 hours, but not delete any of the sessions they have reviewed.

Creating a New Audit Role

If you are the Master Auditor or have been granted the Manage Audit Role right, you can create new audit roles for your organization.

To create a new audit role:

1. Open Audit Manager.
2. Select Audit Roles, right-click, then click **Add Audit Role**.
3. Type a name and description for the new audit role, then click **Next**.
4. Select the type of session.

   For example, select Windows session to limit this audit role to sessions captured by the Agent for Windows.
5. Click **Add** to select additional criteria, such as time constraints, review status, or application used.

   After you click Add, select an attribute and the appropriate criteria, then click **OK**. For example, if you select Time, you can then select specific date range or a period of time, such as the past 24 hours or this year.
6. Click **Execute Query** to test the criteria you have selected by examining the results the query returns.
7. Click **Close** to close the query results, then click **Next**.
8. Select the rights to allow for this role, then click **Next**.
9. Review your settings for this role, then click **Next**.

   By default, the Assign Users and Groups to the Audit Role option is selected so that you can immediately begin populating the new audit role.
10. Click **Finish** to begin adding users and groups to the role.

Assigning Users and Groups to an Audit Role

If you selected the Assign Users and Groups to the Audit Role option at the end of the Add Audit Role wizard, the Assign Users and Groups to the Audit Role wizard opens automatically. You can also open the wizard at any time by right-clicking a specific audit role name in the Audit Manager console and choosing Assign Users and Groups.

To assign users and groups to an audit role:

1. Open Audit Manager.
2. Expand Audit Roles, and select a specific audit role name.
3. Right-click, then click **Assign Users and Groups**.
4. Type all or part of a name and click **OK**.

   If there is more than one name that matches the criteria you specify, select the appropriate name from the names found, then click **OK**. A user or group can be a member of more than one audit role.

Delegating Audit-related Permissions

As the Master Auditor, you can delegate administrative tasks to other Active Directory users or groups. When you grant administrative rights to designated users and groups, you make them “trustees” with permission to perform specific operations. Because delegating administrative tasks to other users is a key part of managing an installation, it is covered in the next chapter.

However, one of the permissions you can delegate to other users and groups is the Manage Audit Role permission. With this permission, selected trustees can create, modify, and delete audit roles. For more information about delegating administrative tasks, see Setting administrative permissions.

Modifying an Audit Role's Properties

The Master Auditor and the audit roles you define are listed under Audit Roles in the Audit Manager console. Selecting a specific audit role name displays a list of members in the right pane. If you are the Master Auditor or been granted the Manage Audit Role permission, you can modify the properties for an audit role after you have created it by selecting the role in Audit Manager, right-clicking, then selecting Properties. For example, you can change the name or description of an audit role, specify the type of sessions members of the role can access, the privileges the audit role grants, and the users and groups who are assigned to the audit role.

How Access Roles and Audit Roles Differ

Depending on whether you have enabled audit and monitoring service together with identity and privilege manager on an agent-managed computer, you might have two sets of roles or just one set of roles and the information captured and the activity allowed depends on the type of role being used.

Identity and Privilege Management Only

If you have only enabled identity and privilege management on a computer and defined access roles:

- Users will not be able to log on if they are assigned to a role where is audit and monitoring service required.
- Users will be able to log on if they are assigned to a role where the audit if possible option is set. In this case, only identity and privilege management audit trail events are captured. For example, the agent records successful and failed logons and when users change from one role to another. Because audit and monitoring service is not enabled, the agent does not capture a video record of all user activity. You also are not able to define audit roles to control who can read or delete audit trail records.
- Users will be able to log on if they are assigned to a role that does not require audit and monitoring service. In this case, only identity and privilege management audit trail events are captured.
- Auditors will not be able to review user activity on these computers. You also are not able to define audit roles to control who can read or delete audit trail records.

If no audit and monitoring service components are installed, you must use the Windows Event Viewer to search for and review audit trail events.

Auditing Only

If you have enabled only audit and monitoring service on a computer and defined access roles:

- Users will be able to log on if they are assigned to a role where audit and monitoring service is required as long as the agent is running.
- Users will be able to log on if they are assigned to a role where the audit if possible option is set. In this case, logging on starts a video record of all user activity on the computer. Because identity and privilege management are not enabled, the user cannot select any access roles that provide desktop, application, or network access rights. The user cannot change roles so only the audit trail records successful and failed logons events.
- Users will be able to log on if they are assigned to a role that does not require audit and monitoring service. In this case, audit trail events are recorded, but no session activity is captured.
- Auditors will be able to review all or selected user activity on these computers, and you can define audit roles to control who has access to the captured user sessions based on the criteria you specify.

Identity and Privilege Management and Auditing on the Same Computer

If you have enabled audit and monitoring service together with identity and privilege management on the same computer and defined access and audit roles:

- Users will be able to log on if they are assigned to a role where audit and monitoring service is required as long as the agent is running. If the agent is stopped for any reason, the user will be allowed to log on only if also assigned a role with a rescue system right.
- Users will be able to log on if they are assigned to a role where the audit if possible option is set. If the audit and monitoring service service is active and you have enabled video capture auditing, both audit trail events and user activity are captured. For example, the agent records successful and failed logons and user activity when users change from one role to another. If the audit and monitoring service service is not enabled or not currently active, the agent does not capture a video record of all user activity.
- Users will be able to log on if they are assigned to a role that does not require audit and monitoring service. In this case, only audit trail events are captured.
- Auditors will be able to review user activity associated with specific roles on these computers, and you can define audit roles to control who has access to the captured user sessions based on the criteria you specify.

This chapter describes how to secure and manage the audit and monitoring service infrastructure after the initial deployment of Delinea software on Windows computers. It includes tasks that are done by users assigned the Master Auditor role for an installation and users who are Microsoft SQL Server database administrators.

Securing an Installation

For production deployments, you can take the following steps to secure an audit and monitoring service installation:

- Use the Installation group policy to specify which installation agents and collectors are part of. By enabling the Installation group policy you can prevent local administrators from configuring a computer to be part of an unauthorized installation.
- Configure a trusted group of collectors to prevent a hacker from creating a rogue collector to collect data from agents.
- Configure a trusted group of agents to prevent a hacker from performing a Denial of Service attack on the collector and database by flooding a collector with false audit data.
- Encrypt all data sent from the collector to the database.

Before you can follow these steps to secure an installation, you must have access to an Active Directory user account with permission to create Active Directory security groups, enable group policies, and edit Group Policy Objects.

To secure an installation using Windows group policy:

1. Open the Group Policy Management console.
2. Expand the forest and domain to select the Default Domain Policy object.
3. Right-click, then click **Edit** to open Group Policy Management Editor.
4. Expand **Computer Configuration** › **Policies** › **CentrifyDirectAudit <!---TODO update filename---> Settings**, then select **Common Settings**.
5. Double-click the **Installation** policy in the right pane.
6. On the Policy tab, select **Enabled**.
7. Click **Browse** to select the installation you want to secure, then click **OK**.
8. Click **OK** to close the Installation properties.

Securing an Audit Store with Trusted Collectors and Agents

By default, audit stores are configured to trust all audited computers and collectors in the installation. Trusting all computers by default makes it easier to deploy and test audit and monitoring service in an evaluation or demonstration environment. For a production environment, however, you should secure the audit store by explicitly defining the computers the audit store can trust.

You can define two lists of trusted computers:

- Audited computers that can be trusted.
- Collector computers that can be trusted.

To secure an audit store:

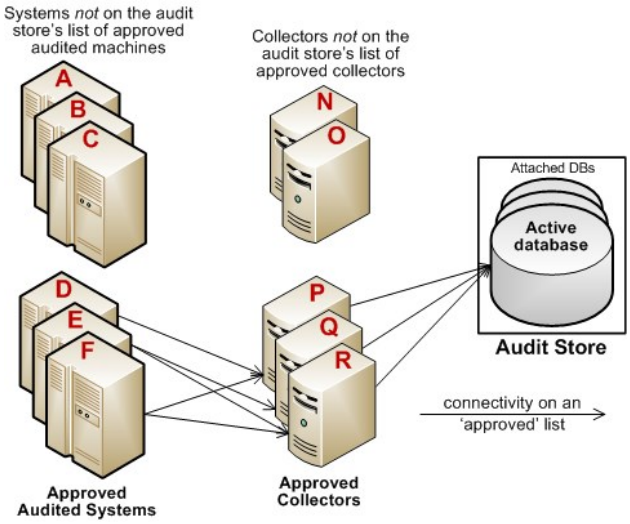
1. Open the Audit Manager console.
2. Expand the installation and Audit Stores nodes.
3. Select the audit store you want to secure, right-click, then select **Properties**.
4. Click the **Advanced** tab.
5. Select **Define trusted Collector list**, then click **Add**.
6. Select a domain, click **OK**, then search for and select the collectors to trust and click **OK** to add the selected computers to the list.

   Only the collectors you add to the trusted list are allowed to connect to the audit store database. All other collectors are considered untrusted and cannot write to the audit store database.
7. Select **Define trusted Audited System list**, then click **Add**.
8. Select a domain, click **OK**, then search for and select the audited computers to trust and click **OK** to add the selected computers to the list.

Only the audited computers you add to the trusted list are allowed to connect to the trusted collectors. All other computers are considered untrusted and cannot send audit data to trusted collectors.

9. Click **OK** to close the audit store properties dialog box.

The following example illustrates the configuration of trusted collectors and trusted audited computers.



In this example, the audit store trusts the computers represented by P, Q, and R.Those are the only computers that have been identified as trusted collectors in the audit store Properties list. The audit store has been configured to trust the audited computers represented by D, E, and F. As a result of this configuration:

- Audited computers D, E, and F only send audit data to the trusted collectors P, Q, and R.
- Trusted collectors P, Q, and R only accept audit data from the trusted audited computers D, E, and F.
- The audit store database only accepts data for its trusted collectors P, Q, and R, and therefore only stores audit data that originated on the trusted audited computers D, E, and F.

Disabling a trusted list

After you have added trusted collectors and audited computers to these lists, you can disable either one or both lists at any time to remove the security restrictions. For example, if you decide to allow audit and monitoring service data from all audited computers, you can open the audit store properties, click the Advanced tab, and deselect the **Define trusted Audited System list** option. You don't have to remove any computers from the list. The audit store continues to only accept data from trusted collectors.

Using security groups to define trusted computers

You can use Active Directory security groups to manage trusted computer accounts. For example, if you create a group for trusted audited computers and a group for trusted collectors, you can use those groups to define the list of trusted collectors and audited computers for the audit store. Any time you add a new computer to one of those groups, thereafter, it is automatically trusted, without requiring any update to the audit store properties.

Securing Network Traffic with Encryption

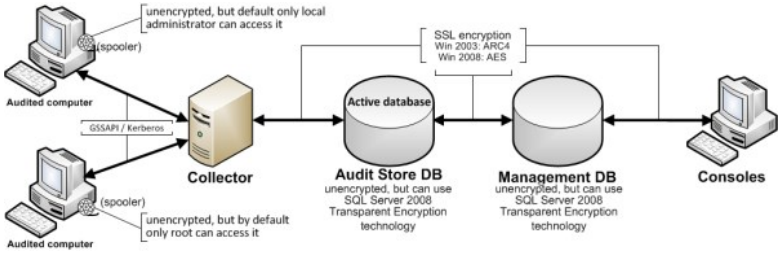
The last step in securing an installation is to secure the data collected and stored through encryption. The following summarizes how data is secured as it moves from component to component:

- Between an audited computer and the spooler that stores the data locally when no collectors are available, audit data is not encrypted. Only the local Administrator account can access the data by default.
- Between the audited computer's data collection service (wdad) and the collector, data is secured using Generic Security Services Application Program Interface (GSSAPI) with Kerberos encryption.
- Between the collector and the audit store database, data can be secured using Secure Socket Layer (SSL) connections and ARC4 (Windows 2003) or AES (Windows 2008) encryption if the database is configured to use SSL connections.
- Between the audit store and management databases, data can be secured using Secure Socket Layer (SSL) connections and ARC4 (Windows 2003) or

AES (Windows 2008) encryption if the database is configured to use SSL connections.

- Between the management database and the Audit Manager console, data can be secured using Secure Socket Layer (SSL) connections and ARC4 (Windows 2003) or AES (Windows 2008) encryption if the database is configured to use SSL connections.

The following illustration summarizes the flow of data and how network traffic is secured from one component to the next.



Enabling Secure Socket Layer (SSL) communication

Although the database connections can be secured using SSL, you must configure SSL support for Microsoft SQL Server as part of SQL Server administration. You must also have valid certificates installed on clients and the database server. If you are not the database administrator, you should contact the database administrator to determine whether encryption has been enabled and appropriate certificates have been installed. For more information about enabling SSL encryption for SQL Server and installing the required certificates, see the following Microsoft support article:

http://support.microsoft.com/kb/316898

Enabling encryption for Microsoft SQL Server Express

If you use Microsoft SQL Server Express, encryption is turned off by default. To secure the data transferred to the database server, you should turn encryption on.

To enable encryption for each audit store and management database:

1. Log on to the computer hosting an audit store or management database with an account that has database administrator authority.
2. Open **SQL Server Configuration Manager**.
3. Select the SQL Server Network Configuration node, right-click **Protocols for** *DBINSTANCE*, then select **Properties**.
4. On the **Flags** tab, select **Yes** for the **Force Encryption** option, then click **OK** to save the setting.

Using a service account for Microsoft SQL Server

When you install Microsoft SQL Server, you specify whether to use Windows authentication or a mix of Windows and SQL Server authentication. You also specify the accounts that the database services should use. By default, system accounts are used. If SQL Server uses a domain user account instead of a system account, you should ensure that the account has permission to update the SQL Server computer object in Active Directory. If the account has permission to update the computer where SQL Server is running, SQL Server can publish its service principal name (SPN) automatically. Getting the correct service principal name is important because Windows authentication relies on the SPN to find services and DirectManage Audit uses Windows authentication for console-to-audit management database connections. If the SPN is not found, the connection between the console and audit management database fails.

The audit management database-to-audit store connection and the collector-to-audit store connection can use either Windows authentication or SQL Server authentication. If SQL Server authentication is used, it does not matter whether the SQL Server instance uses a system account or a service account. If you have configured SQL Server to use Windows authentication only, be sure that the Windows account is allowed to connect to the audit management database and to the audit store database.

Setting Administrative Permissions

When you create a new installation, you become the primary administrator for that installation. As the primary administrator and Master Auditor, you have full control over the entire installation and the ability to delegate administrative tasks to any other Active Directory user or group. When you grant administrative rights to designated users and groups, you make them "trustees" with permission to perform specific operations. You can set granular permissions to tightly control what specific users can do or grant broad authority over operations in an installation.

If you have a large or widely-distributed installation, you can also install additional Audit Manager and Audit Analyzer consoles for the users who have been delegated administrative tasks to use.

To delegate administrative tasks to other users:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Security** tab to delegate administrative tasks for the entire installation.
4. Click **Add** to add Active Directory users or groups to the list of trustees who granted any type of rights on this installation.
5. Select a user or group listed, then select the appropriate rights for that trustee, then click **OK**.

   The following table lists the rights available.

Full Control	All operations on the selected installation.
Change Permissions	Add or remove users and groups as trustees for the installation. Modify permissions for trustees on the selected installation.
Modify Name	Modify display name for the selected installation.
Manage Management Database List	Add or remove management databases for the selected installation.
Manage Audit Store List	Add or remove audit stores for the selected installation.
Manage Collectors	Enable a trusted group of collectors for this audit store. Add a collector to the trusted group of collector in this audit store. Remove collector from the trusted collectors in this audit store. Remove disconnected collector records from this audit store.
Manage Audited Systems	Enable trusted group of audited computers for this audit store. Add a computer to the trusted group of audited computers in this audit store. Remove a computer from the trusted group of audited computers in this audit store. Remove disconnected audited computer records from this audit store.
Manage Audit Role	Add, modify, or remove audit roles in the selected installation. Assign users and groups to audit roles. Remove users and groups from roles.
Manage Queries	Add, modify, or remove queries in the selected installation.
Manage Publications	Add or remove publication locations for the selected installation.
Manage Licenses	Add or remove license keys for the selected installation.
Modify Notification	Enable or disable audit notification in the selected installation. Select the notification message. Select a banner image.
Modify Audit Options	Enable or disable the option to capture video of all user activity on audited computers. Control whether users are allowed to update the review status of their own sessions. Control whether users are allowed to delete their own sessions.
View	Enable to view audited computers and sessions.

1. Click **OK** to complete the delegation of administrative rights for the selected installation.

You can also delegate administrative tasks for individual audit stores and management databases, and set permissions on audit roles. For information about delegating administrative tasks for audit stores, see Configuring permissions for an audit store. For information about delegating administrative tasks for management databases, see Configuring permissions for the management database.

For information about setting permissions on audit roles, see Managing audit roles and auditors.

Managing Audit Stores

An audit store defines a set of Active Directory sites or subnets and a collection of databases that contain audit data. Typically, an installation has one audit store with multiple databases. However, you can add audit stores if you are auditing computers in a large and widely distributed network or have multiple Active Directory sites with computers you want to audit.

Configuring the Scope of an Audit Store

In most organizations, a single audit store is used to map to an Active Directory site. However, there are situations where you might want to define the scope of an audit store based on subnets. For example:

- If you have a subnet that Active Directory considers part of a site that is connected over a slow link you might want to configure a separate audit store and collectors that service audited computers in the remote subnet.
- If you have very large Active Directory site, you might require multiple audit stores for load distribution. You can accomplish this by partitioning an Active Directory site into multiple audit stores based on subnets. Each subnet has its own audit store, set of collectors, and audited computers.

You can configure the scope of an audit store by adding or removing Active Directory sites or subnets.

To configure the scope for an audit store:

1. Open Audit Manager.
2. Expand the installation node, then expand Audit Stores and select a specific audit store name.
3. Right-click, then select **Properties**.
4. Click the **Scope** tab.
5. Click **Add Site** to select an Active Directory site from the list of sites found or click **Add Subnet** to type a specific subnet address and mask.

Configuring Permissions for an Audit Store

If you are the Master Auditor or have Change Permission rights, you can modify the rights granted to Active Directory users or groups. When you enable rights for designated users and groups, you make them "trustees" with permission to perform specific operations.

To configure permissions for managing the audit store:

1. Open Audit Manager.
2. Expand the installation node, then expand Audit Stores and select a specific audit store name.
3. Right-click, then select **Properties**.
4. Click the **Security** tab.
5. Click **Add** to add Active Directory users or groups to the list of trustees who granted any type of rights on this audit store.
6. Select a user or group listed, then select the appropriate rights for that trustee, then click **OK**.

   The following table lists the rights available.

Full Control	All operations on the audit store.
Change Permissions	Modify permissions on this audit store.

Modify Name	Modify display name for this audit store.
Manage Scopes	Add a subnet or Active Directory site to the audit store. Remove a subnet or Active Directory site from the audit store.
Manage SQL Logins	Set the allowed incoming collectors for this audit store's databases. Set the allowed incoming management databases for this audit store's databases.
Manage Collectors	Enable a trusted group of collectors for this audit store. Add a collector to the trusted group of collector in this audit store. Remove collector from the trusted collectors in this audit store. Remove disconnected collector records from this audit store.
Manage Audited Systems	Enable trusted group of audited computers for this audit store. Add a computer to the trusted group of audited computers in this audit store. Remove a computer from the trusted group of audited computers in this audit store. Remove disconnected audited computer records from this audit store.
Manage Databases	Add audit store databases to this audit store. Attach audit store databases to this audit store. Detach an audit store database from this audit store. Change the active database in this audit store. Modify the display name of an audit store database.
Manage Database Trace	Enable or disable database trace. Export database trace.

Managing Audit Store Databases

During the initial deployment, your installation only has one audit store database. As you begin collecting audit data, however, that database can quickly increase in size and degrade performance. Over time, an installation typically requires several Microsoft SQL Server databases to store the data being captured and historical records of session activity, login and role change events, and other information. As part of managing an installation, you must manage these databases to prevent overloading any one database and to avoid corrupting or losing data that you want to keep.

One of the biggest challenges in preparing and managing Microsoft SQL Server databases for storing audit data is that it is difficult to estimate the level of activity and how much data will need to be stored. There are several factors to consider that affect how you configure Microsoft SQL Server databases for auditing data, including the recovery method, memory allocation, and your backup and archiving policies.

For more complete information about managing and configuring SQL Server, however, you should refer to your Microsoft SQL Server documentation.

Selecting a Recovery Model

Standard backup and restore procedures come in three recovery models:

- **Simple**—The simple recovery model allows high-performance bulk copy operations, minimizes the disk space required, and requires the least administration. The simple recovery model does not provide transaction log backups, so you can only recover data to the point of the most recent full or differential backup. The default recovery model is simple, but is not appropriate in cases where the loss of recent changes is not acceptable.
- **Full**—The full recovery model has no work-loss exposure, limits log loss to changes since the most recent log backup, and provides recovery to an arbitrary time point. However, the full recovery model uses much more disk space.
- **Bulk-logged**—The bulk-logged recovery model provides higher performance and minimizes the log space used by disk-intensive operations, such as create index or bulk copy. With the bulk-logged recovery model, you can only recover data to the point of the most recent full or differential backup. However, because most databases undergo periods of bulk loading or index creation, you can switch between bulk-logged and full recovery models to minimize the disk space used to log bulk operations.

When a database is created, it has the same recovery model as the **model** database. Although the simple recovery model is the default, the full and bulk-logged recovery models provide the greatest protection for data, and the full recovery model provides the most flexibility for recovering databases to an earlier point in time. To change the recovery model for a database, use the ALTER DATABASE statement with a RECOVERY clause.

Regardless of the recovery model you choose, you should keep in mind that backup, restore, and archive operations involve heavy disk I/O activity. You should schedule these operations to take place in off-peak hours. If you use the simple recovery model, you should set the backup schedule long enough to prevent

backup operations from affecting production work, but short enough to prevent the loss of significant amounts of data.

Configuring the Maximum Memory for Audit Store Databases

Because Microsoft SQL Server uses physical memory to hold database information for fast query results, you should use a dedicated instance to store auditing data. Because SQL Server dynamically acquires memory whenever it needs it until it reaches the maximum server memory you have configured, you should set constraints on how much physical memory it should be allowed to consume.

The maximum server memory (max server memory) setting controls the maximum amount of physical memory that can be consumed by the Microsoft SQL Server buffer pool. The default value for this setting is such a high number that the default maximum server memory is virtually unlimited. Because of this default value, SQL Server will try to consume as much memory as possible to improve query performance by caching data in memory.

Processes that run outside SQL Server, such as operating system processes, thread stacks, socket connections and Common Language Runtime (CLR) stored procedures are not allowed to use the memory allocated to the Microsoft SQL Server buffer pool. Because those other processes can only use the remaining available memory, they might not have enough physical memory to perform their operations. In most casts, the lack of physical memory forces the operating system to read and write to disk frequently and reduces overall performance.

To prevent Microsoft SQL Server from consuming too much memory, you can use the following formula to determine the recommended maximum server memory:

- Reserve 4GB from the first 16GB of RAM and then 1GB from each additional 8GB of RAM for the operating system and other applications.
- Configure the remaining memory as the maximum server memory allocated for the Microsoft SQL Server buffer pool.

For example, if the computer hosting the Microsoft SQL Server instance has 32GB of total physical memory, you would reserve 4 GB (from first 16 GB) + 1GB (from next 8 GB) + 1 GB (from next 8 GB) for the operating system, then set the Maximum server memory for Microsoft SQL server to 26 GB (32 GB – 4 GB – 1 GB – 1 GB = 26).

For more information about how to configure Microsoft SQL Server maximum memory setting and other memory options, see the following Microsoft article: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms178067(v=sql.105)

You should configure the maximum memory allowed for the Microsoft SQL Server instances hosting audit store databases and the management database. However, this setting is especially important to configure on the Microsoft SQL Server instance hosting the active audit store database.

Using Transact-SQL to Configure Minimum and Maximum Memory

You can control the minimum and maximum memory that the SQL Server buffer manager uses by issuing Transact-SQL commands. For example:

```
sp_configure 'show advanced options', 1
reconfigure
go
sp_configure 'min server memory', 60
reconfigure
go
sp_configure 'max server memory', 100
reconfigure
go
```

For more information about configuring SQL Server and setting minimum and maximum server memory using T-SQL, see https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/server-memory-server-configuration-options?view=sql-server-2017

Estimating Database Requirements Based on the Data you Collect

To determine how audit and monitoring service will affect database capacity, you should monitor a pilot deployment of 20 to 25 agents with representative activity to see how much data is produced daily. For example, some audited computers might have few interactive user sessions or only short periods of activity. Other audited computers might have many interactive user sessions or long sessions of activity on average.

During the pilot deployment, you want to the following information:

- How many interactive user sessions occur daily on each computer?
- How long do sessions last on average?
- What are the activities being captured, and what is the average size of each session being captured?
- How long do you need to store the captured data to balance performance and storage?

- What is the data retention period for audited data?

From the information you collect in the pilot deployment and the data retention policy for your organization, you can estimate the database size using the following guideline:

(number of agents) x (number of sessions per agent) x (average data size per session) x (retention days)

Results in the estimated size of the Microsoft SQL Server database for the number of days in the retention policy

For example, if an average session generated 100 KB in the database and the installation had 250 agents, 10 sessions per agent, and a six-month retention period (about 130 working days), the storage requirement for the audit store database would be 36.9 GB:

250 agents x 10 sessions/agent each day x 100 KB/session x 130 days = 32,500,000 KB

The following table shows examples of the data storage requirement in an installation with Windows agents, typical levels of activity with an average of one session per day on each audited computer, and the recovery mode set to Simple:

100	20 minutes	806 KB - low activity	79 MB	394 MB	10 GB
50	25 minutes	11.56 MB - high activity	578 MB	2.81 GB	73.36 GB
100	20 minutes	9.05 MB - high activity	905 MB	4.42 GB	115 GB

In this example, an installation with 100 Windows agents with low activity would require approximately 10 GB for the audit store database to keep audit data for 6 months. An increase in the number of interactive sessions, session length, or average session size would increase the database storage required.

If SQL Server requires more space to accommodate the new data, it expands the database file immediately, which can cause degraded performance. To reduce the effect of database expansion on performance, allocate sufficient space to support database growth. In addition, monitor database space and when space is low, schedule a database expand operation for an off-peak time.

Adding New Audit Store Databases to an Installation

When you first set up an installation, you also create the first audit store and audit store database. By default, that first database is the active database. As you begin collecting audit data, you might want to add databases to the audit store to support a rolling data retention policy and to prevent any one database from becoming a bottleneck and degrading performance.

Only one database can be the active database in an audit store at any given time. The computer hosting the active database should be optimized for read/write performance. As you add databases, you can change the older database from active to attached. Attached databases are only used for querying stored information and can use lower cost storage options.

Note: {/b}A single instance of Microsoft SQL Server can host multiple databases.

Audit store databases have the following characteristics:

- A database can be active, attached, or detached.
- Only one database can be actively receiving audit data from collectors.
- A database cannot be detached while it is the active database.
- A database that was previously the active database cannot again be the active database.
- If a detached database contains parts of sessions presented to the Audit Analyzer, a warning is displayed when the auditor replays those sessions.

Rotating the Active Database

Database rotation is a management policy to help you control the size of the audit store database and the performance of database operations. There are several reasons to do database rotation:

- It is more difficult to manage one large database than multiple small databases.
- Performance is better with multiple small databases.
- Backing up, restoring, archiving, and deleting data all take significantly more time if you work with one large database.

- Database operations take very little time when you work with multiple small databases.

For audit and monitoring service, you can implement a database rotation policy by having the collector write data to a new database after a certain period of time. For example, the collector in site A writes data to the database siteA-2014-11 in November, then write data to database siteA-2014-12 in December and to the database siteA-2015-01 in January. By rotating from one active database to another, each database stays more compact and manageable.

Creating a New Database for Rotation

You can rotate from one active database to another at any time using the Audit Manager console.

To create a new database for rotation:

1. Open Audit Manager.
2. Expand the installation node, then expand Audit Stores and a specific audit store name.
3. Select Databases, right-click, then select **Add Audit Store Database** to create a new database.
4. Select the **Set as Active database** option so collectors start writing to the newly created database.

It is possible to write a script to automate the database rotation process. For details, see the SDK documentation.

Database Archiving

To implement periodic archiving, add a new active database, leave one or more previous databases attached, and take the oldest database off-line for archiving.

Queries During Rotation and Archiving

If the database backup program supports online backup, the Audit Analyzer can still query the database while the backup is in progress. However, the backup program may block updates to the session review status. If the backup program does not support online backup, the database will be offline until the backup is complete.

Database Backups

You can back up a database whether it is attached to the audit store or detached from the audit store.

Allowed Incoming Accounts

You can specify the accounts that are allowed to access the audit store database. By configuring these accounts, you can control which collector computers can connect to the audit store database and which management databases have access to the data stored in the audit store database.

Your account must have Manage SQL Login permission to configure the incoming accounts.

To configure allowed accounts:

1. Open Audit Manager.
2. Expand the installation node, then expand Audit Stores and select a specific audit store name.
3. Select a database under the audit store, right-click, then select **Properties**.
4. Click the **Advanced** tab.
5. Click **Add** to add a collector or management database account. For example:

   ![alt](images/8d96b4edb390e2b9de495a3a7399d708.png "Add Collector or Management Database Account)

6. Select an authentication type.
   - If you select Windows authentication, you can browse to select a computer, user, or group to add.
   - If you select SQL Server authentication, you can select an existing SQL Server login or create a new login.

Connections should use Windows authentication whenever possible. However, computers in an untrusted forest cannot connect to an audit management database using Windows authentication. To allow connections from an untrusted forest, add a SQL Server login account as the incoming account for the management database.

Managing the Management Database

The audit management database keeps track of where components are installed and information about the installation. To connect to the database or manage its properties, select a specific installation name in Audit Manager, right-click, then select **Management Databases**.

Configuring the Scope of the Management Database

The audit management database stores information about the set of Active Directory sites or subnets it supports. You can modify the scope of the management database if you are auditing computers in a large and widely distributed network or have multiple Active Directory sites with computers you want to audit.

To configure the scope for a management database:

1. Open Audit Manager.
2. Select the installation name, right-click, then select **Management Database**.
3. Click **Properties**, then click the **Scope** tab.
4. Click **Add Site** to select an Active Directory site from the list of sites found or click **Add Subnet** to type a specific subnet address and mask.

Configuring Permissions for the Management Database

If you are the Master Auditor or have Change Permission rights, you can modify the rights granted to Active Directory users or groups. When you enable rights for designated users and groups, you make them "trustees" with permission to perform specific operations.

To configure audit store security:

1. Open Audit Manager.
2. Select the installation name, right-click, then select **Management Database**.
3. Click **Properties**.
4. Click the **Security** tab.
5. Click **Add** to add Active Directory users or groups to the list of trustees who granted any type of rights on this management database.
6. Select a user or group listed, then select the appropriate rights for that trustee, then click **OK**.

   The following table lists the rights available.

Full Control	All operations on the management database.
Change Permissions	Modify permissions on the management database.
Modify Name	Modify display name for this management database.
Manage Scopes	Add a subnet or Active Directory site to the management database. Remove a subnet or Active Directory site from the management database.
Manage SQL Logins	Set the allowed incoming accounts for the management database. *Database owner is by definition an allowed user.* Set the outgoing account for the management database.
Remove Database	Remove this audit management database from the installation.
Manage Database Trace	Enable or disable database trace. Export database trace.

Managing Collectors

You can view information about the collectors you have deployed in the Audit Manager console. For example, for each collector, you can see the location of the collector on the network, whether the collector is connected to or disconnected from the audit store, and how long a connected collector has been running since it was last restarted, the audit store to which the collector is assigned, and the active database to which the collector is currently sending audit data. You can also see the audited computers that currently connected to each collector and the audited computers that are not currently connected to this collector.

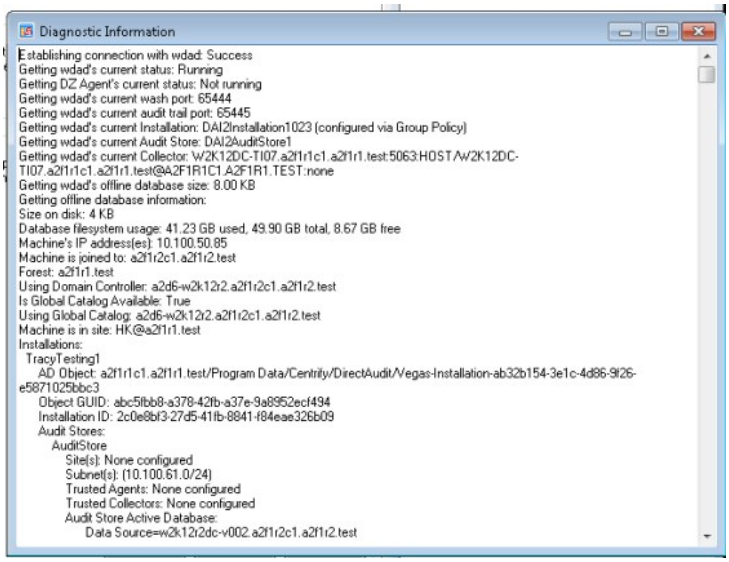
If you install the collector service on a computer but it has never connected to any agents or audit stores, it is not included in collector list on the Audit Manager console.

Monitoring Collector Status Locally

In addition to the information available in the Audit Manager console, the Windows computers on which you have installed a collector provide a local Collector Control Panel applet. The Collector Control Panel displays information about current connectivity and status for the local collector. You can use the control panel to configure the collector port number, installation, and authentication type if you want to make changes after the initial deployment. You can also use the control panel to start, stop, or restart the collector service, and to generate diagnostic information about the collector.

1. Log on to the computer on which you have installed a collector.
2. In the list of applications on the Windows Start menu, click **Audit Collector Control Panel** to open the audit collector control panel.
3. On the General tab, click **Configure** to change the port number, installation, or type of authentication to use when connecting to the audit store.

   The General tab also displays current configuration and status for the local collector service. If you make changes, the new information is displayed after a short period of time.
4. Click **Stop** if you want to temporarily stop a running service, or **Restart** if you want to stop and immediately restart a running collector service.
5. Click the Troubleshooting tab, then click **Diagnostics** to generate diagnostic information about the installation the collector is part of, the Active Directory site or subnets associated with the audit store the collector connects to, the collector status, and other information. For example:



   After you generate diagnostic information, you can right-click to select all of the text. With the text selected, right-click, and select Copy to copy and paste the diagnostic report into a text file.
6. Click **Options** to specify the level of detail to include in the log file or to turn off logging.

   The default log level reports informational messages, warnings, and errors. You can click **View Log** to see information in the current log file.
7. Click **Close** to return to the agent configuration panel.

Removing Collectors

If you want to remove a collector, you can use the Programs and Features > **Uninstall a program** control panel or the setup program you used to install the collector.

If you run the setup program, select the collector from the list of components, then click Next. Because a collector is installed, the wizard prompts you the Change, Repair or Remove the collector. Click **Remove**.

Managing Audited Computers and Agents

You can see information about audited computers and the audit and monitoring service status of Agents for Windows using the Audit Manager console. For example, for each audited computer, you can see the computer name and IP address, whether the audited agent is currently connected or disconnected, and how long the agent has been running since it was last restarted. You can also see the collector to which the agent is sending data and the audit store and audit store database where the audit data is stored.

Monitoring Agent Status Locally

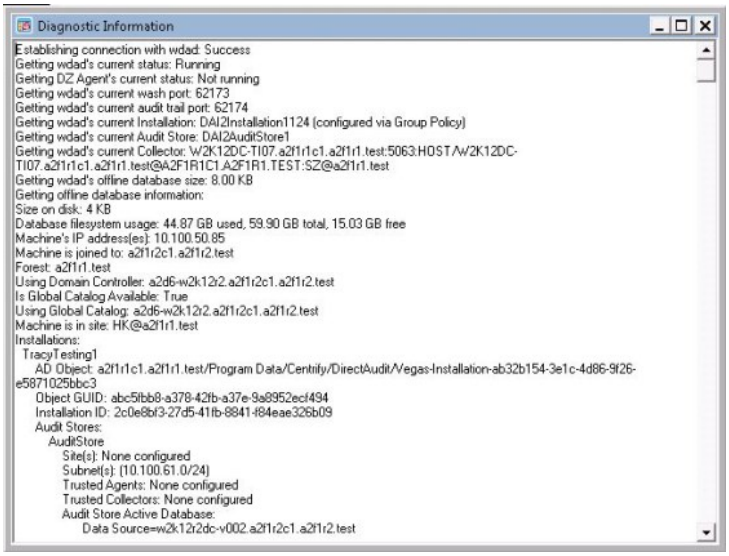
In addition to the information available in the Audit Manager console, the Windows computers on which you have installed a Agent for Windows with audit and monitoring service enabled include a local agent configuration panel applet. The agent configuration panel displays information about current connectivity and status for the local agent. You can use the agent configuration panel to configure the color depth, offline storage, or installation if you want to make changes after the initial deployment. You can also use the agent configuration panel to generate diagnostic information about the agent.

To use the agent configuration panel:

1. Log on to the computer on which you have installed a Agent for Windows with audit and monitoring service enabled.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Click **Auditing and Monitoring Service**.
4. Click **Settings**.
5. On the General tab, click **Configure** to change the color depth, offline storage file location and maximum size, and the installation to use for the local agent.

   > **Note**: The offline storage location should be an empty folder. If you select a folder that contains any files other than the spooled audit data, those files may be moved or lost.

   The General tab also displays current configuration and status for the local agent. If you make changes to the configuration, the new information is displayed after a short period of time. If the agent cannot connect to any collector, it spools audit data to the offline data location. When it finds a collector, the agent sends the spooled data to it. The offline storage space is not reclaimed until all of the spooled data has been sent to a collector.
6. Click the Troubleshooting tab, then click **Diagnostics** to generate diagnostic information about the installation the agent is part of, the collector the agent sends data to, the size of offline storage, and other information. For example:



After you generate diagnostic information, you can right-click to select all of the text. With the text selected, right-click, and select Copy to copy and paste the diagnostic report into a text file.

7. Click **Options** to specify the level of detail to include in the log file or to turn off logging.

   The default log level reports informational messages, warnings, and errors. You can click **View Log** to see information in the current log file.

8. Click **Close** to return to the agent configuration panel.

Setting the Color Depth for Captured Sessions

Because audit and monitoring service captures user activity as video, you can configure the color depth of the sessions to control the size of data that must be transferred over the network and stored in the database. A higher color depth also increases the CPU overhead on audited computers but improves resolution when the session is played back. A lower color depth decreases the amount of data sent across the network and stored in the database. In most cases, the recommended color depth is medium (16 bit). The CPU and storage estimates in this guide are based on a medium (16 bit) color depth.

To change the color depth for captured sessions:

1. Log on to the computer where the Agent for Windows is installed.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Click **Auditing and Monitoring Service**.
4. Click **Settings**.
5. On the General tab, click **Configure**
6. Select the maximum color quality for recorded sessions, then click **Next**.
7. Follow the prompts displayed to change any other configuration settings.

Removing an Audited Computer

If an audited computer has been removed from the installation, the audited computer will continue to be listed on the Audit Manager console as Disconnected. To remove the decommissioned audited computer, select Delete from its context menu.

Adding an Installation

Although a single installation is the most common deployment scenario, you can configure multiple installations. For example, you can use separate installations to provide concurrent production and test-bed deployments or to support multiple administrative domains within your organization.

To create a new installation:

1. Open Audit Manager.

2. Select the root node, right-click, then select **New Installation**.

3. Follow the prompts displayed.

   The steps are the same as the first installation. For more information, see Creating a new installation.

4. Choose the appropriate installation for each collector using the Collector Configuration wizard.

5. Choose the appropriate installation for each agent using the Agent Configuration wizard.

Delegating Administrative Tasks for a New Installation

The account you use to create a new installation is the default administrator and Master Auditor with full control over the entire installation and the ability to delegate administration tasks to other Active Directory users or groups. You can grant permission to perform administrative tasks to other users by opening the Properties for each component, then clicking the Security tab.

Opening an Installation in a New Console

If you create multiple installations at the same site, you can select the installation name, right-click, then select **New Window From Here** to keep consoles for different installations separate from each other. Creating a new window for each installation can help you avoid performing operations on one installation that you intended to perform on another.

Closing an Installation

The Audit Manager console allows you to manage multiple installations. To remove the current installation from the console, but not physically remove the database or the information published to Active Directory, you can select the installation name, right-click, then select **Close**.

Publishing Installation Information

DirectManage Audit publishes installation information to a service connection point (SCP) object in Active Directory so that audited computers and collectors can look up the information. If the published locations for multiple SCPs in the same installation are not the same, or if collectors cannot read from at least one of the published locations, the collectors are unable to determine which audit store is the best match for the sites and subnets, and so they do not attempt to connect to an audit store.

Permission to publish to Active Directory

Only administrators who have been delegated permission to modify various attributes of the installation can publish those attributes to Active Directory.

If you do not have Active Directory permission to modify the installation, the updates are kept in the audit management database, and a message is issued to notify you that the installation information could not be updated in Active Directory.

Synchronizing Installation Information

If you have an Active Directory account with permission to publish information about the installation, you can update the service connection point.

To publish the service connection point for an installation:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Publication** tab, then click **Synchronize** to publish the information.

   In a multi-forest or DMZ environment, this tab lists multiple Active Directory locations to which to publish.

4. Click **OK** to close the installation properties.

Removing or Deleting an Installation

Before you can remove or delete an installation, you must do the following:

- Run the setup program to remove all agents and collectors and collector service connection points (SCPs).
- Detach and remove all audit store databases.

- Open the Installation Properties and click the **Publications** tab to make sure only one installation service connection point (SCP) is listed.

  Note: {/b}To remove service connection points on other sites, contact an administrator with publication permission on those sites.

To remove or delete an installation, select the installation in the Audit Manager console, right-click, then select **Remove** to open the Remove installation dialog box.

- Click **Remove** to remove the installation but *not* delete the management database from the SQL Server instance.
- Click **Delete** to remove the installation and delete the management database from the installation of SQL Server.

  Note:All the publications published to Active Directory are removed when you remove or delete an installation.

Delinea software includes diagnostic tools and log files to help you trace the source of problems if they occur. Diagnostic reports and log files allow you to periodically check for issues and view information about operations on the computers you manage. The information is useful for troubleshooting and in resolving cases with the help of Delinea Support.

This chapter describes how to find log files, set the level of detail recorded in log files, and use diagnostic tools to retrieve information about the operation of the Agent and Server Suite components. This chapter also covers common questions to help you identify and correct problems on the computers you manage.

Solving Problems with Logging On

After you have installed the Agent for Windows and joined the computer to a domain, users cannot log on without a role assignment. The role, however, can be assigned to a local account or a domain account, or the role can be assigned the right to access a remote computer. Consequently, users might encounter problems logging on after the agent is deployed. For example, you might find that users can log on to the computer using a local account but cannot log on using their domain account or have trouble connecting to a remote server.

If users report problems logging on, there are some things you can try to troubleshoot the issue:

- Check the logon rights for the affected users.

  To do this, log on as an administrator and execute dzinfo *user-name* (where *user-name* is the name of the user experiencing problems logging on). You can also check user logon rights using the Authorization Center.

- Try to log on using a local user account or using a different domain account if you have more than one account available.

- Determine whether the computer you are using is connected or disconnected from the network. In rare cases, authorization information might not be available when a computer disconnected from the network.

- If users cannot log on to a remote computer, confirm that they have a role that has the remote logon system right and that the computer itself is configured to allow users to log on remotely. Open the Authorization Center to review the list of roles and their associated rights for any user.

- Check the computer's local security policy or applied group policies to verify whether the user is allowed to log on interactively or through a remote desktop connection. For example, most domain users are not allowed to log on locally on domain controllers.

  Depending on how your organization has configured native Windows security policies, users might need to be members of a specific Windows security group—such as Server Operators or Remote Desktop Users—to log on to specific computers locally or remotely even if they have been granted access rights using the Windows Login role or a custom role definition.

- Check to see whether the computer is in Rescue mode.

  In Rescue mode, access to a computer is granted only to users who have Rescue rights. For information about adding Rescue rights to a role, see System rights allow users to log on. In general, a computer enters Rescue mode because the Windows agent authorization service has stopped. Possible causes include the following:

  - The computer is not connected and the local authorization cache has not been initialized or is corrupt.
  - The local authorization cache cannot be updated because the file system is full.

    See Working with the authorization cache on managed computers for more information about the authorization cache and the conditions under which a computer is considered to be not connected.

Accessing Network Computers with Privileges

Depending on how you have defined the roles users are assigned, it is possible for users to see potentially misleading information in certain applications or be unable to perform the administrative tasks as they expect. For example, if users select a role with administrative privileges to access an application such as SQL Server Configuration Manager or Microsoft SQL Server Management Studio and connect to a remote SQL Server instances, it might appear as if they have permission to start and stop services or perform other tasks. However, if the role does not include network access rights for the remote SQL Server instance, users will not have the appropriate permission to perform those tasks.

You can check whether the selected role includes network access rights using the Authorization Center. If the role being used does not include network access rights, check whether the user has additional network roles available to use in conjunction with the local role. If the role being used includes network

access rights, you should check whether those rights are applicable on the network computer the user is attempting to manage. Users must be assigned to the role that has network access rights on the remote server.

Refreshing Cached Information on Managed Computers

Authorization information is cached on the local computer to improve performance and to allow the use of elevated privileges even if users are disconnected from the network. If you make changes to rights, role definitions, or role assignments, you can refresh the information stored in the cache on managed computers to ensure the agent has the most up-to-date information about current rights and roles. If users are experiencing authorization problems or issues with their access rights (for example, if the management console shows that a user has logon rights, but dzinfo or the authorization center does not show that the user has logon rights), you should try refreshing the cache to make sure any changes you have made take effect.

You can refresh the cache using agent configuration panel or the dzrefresh command line program in a Command Prompt window if you have the appropriate permissions.

Analyzing Information in Active Directory

One important way you can troubleshoot your environment is by running the Analyze command. The Analyze command enables you to selectively check the integrity of information stored in Active Directory. With the Analyze wizard, you can check for a variety of potential problems, such as empty zones, invalid role assignments, or orphaned role assignments.

> **Note**: When you run the Analyze command, only the zones that are open are checked.

To check for problems in the Active Directory forest:

1. Open Access Manager.

   If you are prompted to connect to a forest, specify the forest domain or domain controller to which you want to connect.

2. Select the root node, right-click, then click **Analyze**.

3. Select the types of checks you want to perform, then click **Next** to generate the report.

   You can select All to perform a complete check of the Active Directory forest. However, some of the analysis options are only applicable for Linux and UNIX computers or UNIX user and group profiles. For more information about any analysis option, see the Access Manager help or the *Administrator's Guide for Linux and UNIX*.

4. Review the result summary, then click **Finish**.

5. If the result summary indicates any issues, you can view the details by selecting **Analysis Results** in the console tree and viewing the information listed in the right pane.

6. Select individual warnings or errors, right-click, then select **Properties** for additional information.

Common Scenarios that Generate Errors and Warnings

For most organizations, it is appropriate to check the data integrity of the Active Directory forest on a regular basis. Although running the Analyze command frequently may not be necessary for small networks with few domain controllers, there are several common scenarios that you should consider to determine how often you should check the forest for potential problems.

The most likely reasons for data integrity issues stem from:

- Multiple administrators performing concurrent operations.
- Administrators using different domain controllers to perform a single operation.
- Replication delays that allow duplicate or conflicting information to be saved in Active Directory.
- Insufficient permissions that prevent an operation from being successfully completed.
- Network problems that prevent an operation from being successfully completed.
- Partial or incomplete upgrades that result in inconsistency of the information stored in Active Directory.
- Using scripts or ADSI Edit rather than the console to create, modify, or delete objects in Active Directory, which may lead to corrupted or invalid information.

Running Analyze periodically helps to ensure that the scenarios that can cause problems are reported in the Analysis Results, enabling you to take corrective action.

Responding to Errors and Warnings

Depending on the type of warning or error generated in the Analysis Results, you might be able to take corrective action or access additional information. For example, if a computer account lacks the necessary permission to update Active Directory with the agent version it has currently installed, the Analysis Result will enable you to update the computer's account permissions to allow changes to that attribute.

To review additional information or take corrective action, select the error or warning in the list of Analysis Results after running the Analyze wizard, right-click, then select Properties. For more information about responding to analysis results, see the Access Manager help or the *Administrator's Guide for Linux and UNIX*.

Running Diagnostics and Viewing Logs for the Agent

The Agent for Windows provides logging and diagnostic services. If you have administrative access on a local computer, you can generate diagnostic information about the operation of the Agent for Windows and view and save the current content of the log file from the agent configuration panel. For example, you can generate diagnostic information about user sessions, user roles, desktops, and elevated account access, as well as detailed information about auditing from the agent configuration panel.

There are three different types of diagnostics information available:

- **Audit & Monitoring Service** provides the diagnostic information related to the auditing and monitoring service.
- **Identity Platform** provides the diagnostic information related to Privileged Access Service, such as for MFA. This diagnostics tool runs the following tests:
  - **Agent Service Connectivity Check:** Checks to see if the agent is in service, and if the agent is running in a normal state. Also determines whether the agent is in a zone, or is configured to use zoneless mode.
  - **Connector Connectivity Check:** Determines whether all connectors in the network can be connected properly. whether the certificates (IWA and cloud) have been installed properly. Also determines whether the agent can be connected without a trusted certificate problem.
  - **Identity Platform Connectivity Check:** Determines whether a connection to the cloud tenant is functional. Checks for problems with DNS, the firewall, and proxy server settings.
  - **MFA Configuration Check:** Determines whether the local computer has been configured properly. If the computer is in a zone, the test also checks whether MFA complies with the configuration defined in the zone.
  - **MFA Role and Permission Check:** Verifies whether role permissions are set properly in the Privileged Access Service Admin Portal.
  - **Offline MFA Provisioning Check:** Determines if the computer has been configured with an offline MFA profile or not.
- **Privilege Elevation Service** provides the diagnostic information related to privilege management.

You can view these diagnostics tools either from the Windows system tray or from the agent configuration panel.

To view diagnostics from the Windows system tray:

1. Log on to a computer where the Agent for Windows is installed.
2. In the Windows system tray, right-click the Delinea icon and click **Troubleshooting**, then select the service for which you want to view diagnostic information (your options may vary depending on what services are enabled on the computer):
   - **Audit & Monitoring Service** opens a dialog box with a text-based summary of diagnostic auditing and monitoring information.
   - **Identity Platform** runs a series of connectivity tests and lists out the results of each test.
   - **Privilege Elevation Service** opens a dialog box with a text-based summary of diagnostic privilege elevation information.

To generate diagnostics or view the log file from the agent configuration panel:

1. Log on to a computer where the Agent for Windows is installed.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Select the service for which you want to view information:
   - **Audit & Monitoring Service** opens a dialog box with a text-based summary of diagnostic auditing and monitoring information.
   - **Identity Platform** runs a series of connectivity tests and lists out the results of each test.
   - **Privilege Elevation Service** opens a dialog box with a text-based summary of diagnostic privilege elevation information.
4. Click **Settings**.
5. Click the **Troubleshooting** tab.
6. Click **Diagnostics** to generate diagnostic information.
7. Select the Diagnostic Information displayed, right-click, then select **Copy** to copy and paste the output to a file for further analysis.
8. Click **View Log** to display the current log file for the local agent.
9. Click **Options** to see or change the location of the log file or the level of detail recorded in the log file.

Sample Diagnostic Report

For example, if you are viewing information about the privilege elevation service, the diagnostic report might be similar to this:

Product: Infrastructure Services (Name and Version information)
Computer: DC2008R2-LG
Joined Domain: finsterwald.org
Zone: finsterwald.org/Acme Pubs/Zones/HeadquartersAgent State: Connected
Time: 2017-10-16 12:38:03.620 -08:00
Session information:
Session 1
SAM Name: FINSTERWALD\anton.splieth
Logon Type: Console
Always Audit: Yes
Desktops:
Default
GUID: de1dd94a-b671-4b37-baa4-9b2c1b70e776
DZ Logon Id: (0x0)
Local Role: Self
Network Roles: Self
Always Audit: Yes
Audit Flag: On
UAC Restrictions: No
SQL-DBA
GUID: fccb2382-3800-4f3c-9569-922048f91375
DZ Logon Id: (0x9ba99)
Local Role: SQL-DBA/Headquarters
Network Roles: Self
Always Audit: Yes
Audit Flag: On
UAC Restrictions: No
Network Drives: No

Logon information:
Logon ID (0x9ba99)
Logon GUID: 38407dd1-0165-458e-b45d-686a07e87805
Base Logon ID: (0x77163)
Base SAM Name: FINSTERWALD\anton.splieth
ElevatedAccount: (ElevatedSelfAccount, AdditionalGroups=(count=1, items=(S-1-5-32-544)))
Local Role: SQL-DBA/Headquarters
Network Roles: None
Should Audit: Yes
Logon ID (0x22bfee)
Logon GUID: 1b50b739-461c-410e-803c-ed52d4ba1e80
Base Logon ID: (0x77163)
Base SAM Name: FINSTERWALD\anton.splieth
ElevatedAccount: (ElevatedSelfAccount, AdditionalGroups=(count=1, items=(S-1-5-32-544)))
Local Role: SQL-DBA/Headquarters
Network Roles: None
Should Audit: Yes

Domain last access information:
Forest finsterwald.org: Connected
Domains: finsterwald.org: Connected
Multi-factor Authentication information: None

Done.

Enabling Detailed Logging for Audit and Monitoring Service Components

In addition to the log files for the Agent for Windows, there are log files for other audit and monitoring service components to record information about operations performed by those components on a local computer. If you have audit and monitoring service components installed, you can view the log files or change log file options for those components to assist Delinea Support when troubleshooting issues.

Enabling Detailed Logging for an Audited Computer

If you are troubleshooting an audit and monitoring service-related issue, you should enable detailed logging for the audit and monitoring service service on the computers being audited. For Windows computers, you can enable detailed logging using the agent configuration panel.

To enable detailed logging on an audited computer:

1. Log on to an audited computer.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Click **Audit & Monitoring Service**.
4. Click **Settings**.
5. Click the **Troubleshooting** tab.
6. Click **Options**, change the logging level to **Trace messages**, then click **OK**.
7. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
8. Click **View Log** to view the current log file.

   From the log file window, you can also click **File > Save As** to save the log file.
9. Send an email to Delinea Support with the log file from the location specified in Step 7 as an attachment.
10. Click **Options**, change the logging level back to its default setting of **Informational messages**, then click **OK**.
11. Click **Close** to return to the agent configuration panel.

Enabling Detailed Logging for the Collector Service

If you are troubleshooting an audit and monitoring service-related issue, you should enable detailed logging for the collector service on the computers where the collector service runs.

To enable detailed logging on a collector:

1. In the list of applications on the Windows Start menu, click **Audit Collector Control Panel** to open the audit collector control panel.
2. Click the **Troubleshooting** tab.
3. Click **Options**, change the logging level to **Trace messages**, then click **Apply**.
4. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
5. Click **View Log** to view the current log file.

   From the log file window, you can also click File > Save As to save the log file.
6. Send an email to Delinea Support with the log file from the location specified in Step 4 as an attachment.
7. Click **Options**, change the logging level back to its default setting of **Informational messages**, then click **OK**.
8. Click **Close** to return to the Collector Control Panel.

Enabling Detailed Logging for Audit and Monitoring Service Consoles

In most cases, troubleshooting audit and monitoring service-related issues requires information about the operation of the agent and the collector or database activity. However, in some cases, it might be necessary to capture detailed information about the operation of Audit Manager or Audit Analyzer.

To capture detailed information for Audit Manager or Audit Analyzer:

1. Log on to a computer where the Audit Manager or Audit Analyzer console is installed.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Click **Audit & Monitoring Service**.
4. Click **Settings**.
5. Click the **Troubleshooting** tab.
6. Click **Options**.
7. In the Log Settings tab, change the logging level to **Trace messages**, then click **OK**.
8. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
9. Send an email to Delinea Support with the log file from the location specified in Step 8 as an attachment.
10. Click **Options**, change the logging level back to its default setting of **Warning messages**, then click **OK**.
11. Click **Close** to return to the agent configuration panel.

Enabling Audit and Monitoring Service Performance Counters for the Collector

If you have enabled audit and monitoring service and installed the collector service on a local Windows computer, you can add audit-specific performance counters to Performance Monitor to help you analyze and resolve audit-related issues. When you install the collector, the performance counters are added automatically. When you uninstall the collector, the counters are automatically removed from Performance Monitor.

For more information about troubleshooting in an audit installation, see the *Auditing Administrator's Guide*.

Tracking Database Activity

Database traces are used to help diagnose problems in the management database or audit store databases. For example, database traces can help to identify inconsistencies caused by hardware errors or network interruptions. After you enable database tracing, DirectManage Audit tracks all of the SQL statements and debug messages from the audit management database or audit store, and records the information in the database server.

> **Note**: Tracing database operations affects database performance. You should only activate a database trace if you require this information for troubleshooting. Before you start a database trace, try to reduce the load on the database instance as much as possible, then only perform the actions needed to reproduce the issue you are troubleshooting. Turn off database tracing as soon as you have logged the activity you need for the analysis of database operations. The trace for each database can take up to 800MB of server disk space. After you turn off database tracing, restart the SQL Server instance to reset the disk space.

Starting a Database Trace

You can start a database trace for a management database or an audit store database.

To start database tracing:

1. Open Audit Manager.
2. Select an installation name, right-click, then click **Properties**.
3. Click the **Database Trace** tab.

   This tab displays basic information about the management databases and audit store databases for the selected installation. In the Trace Status column, you can see whether tracing is enabled or disabled for each database.
4. Select a management or audit store database in the list, then click **Enable** to start tracing on the database selected.
5. Click **OK**, then perform the database actions for which you want to capture information.

Stopping the Database Trace

You should turn off database tracing immediately after you have logged the activity you need for the analysis of database operations.

To stop database tracing:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Database Trace** tab.

4. Select the management or audit store database that has tracing enabled, then click **Disable** to stop tracing on the database selected.
5. Click **Export** to save the database trace from the selected databases to a file with comma-separated values (.csv).
6. Follow the prompts displayed in the Export Database Trace wizard to save the information to a file.

Exporting the Database Trace for a Management Database

The Export Database Trace wizard prompts you for different information depending on whether the database trace is for a management database or an audit store database. For example, if you generate a database trace for a management database then click **Export**, the Export Database Trace wizard prompts you for user accounts.

To export the database trace:

1. Select a start date and time for the **From** filter and an end date and time for the **To** filter, then click **Next**.
2. Click **Add** to search for and select users, then click **Next**.

   By default, you can search for users in the entire directory, you can click **Object Types** or **Locations** to change the scope of the search scope, or click **Advanced** specify other criteria.
3. Accept the default folder location or click **Browse** to select a different location, then click **Next**.
4. Review your selections, then click **Next**.

   By default, the wizard save the file as *installation_name*.csv and opens the file location.
5. Click **Finish**, then click **OK** to close the installation properties.

Exporting the Database Trace for Audit Store Databases

When you select an audit store from the lower area of the **Database Trace** tab on the **Properties** page and click the lower **Export** button, the wizard opens with a date/time **Export Criteria** page. On the second page, the wizard asks you to pick the domain and computer.

To export the database trace:

1. Select a start date and time for the **From** filter and an end date and time for the **To** filter, then click **Next**.
2. Click **Add** to search for and select collectors, then click **Next**.

   By default, you can search for computers in the entire directory, you can click **Object Types** or **Locations** to change the scope of the search scope, or click **Advanced** specify other criteria.
3. Click **Add** to search for and select management database computers, then click **Next**.
4. Accept the default folder location or click **Browse** to select a different location, then click **Next**.
5. Review your selections, then click **Next**.

   By default, the wizard save the file as *audit_store_name*.csv and opens the file location.
6. Click **Finish**, then click **OK** to close the installation properties.

Delegating Database Trace Management

You can delegate the authority to manage database tracing by granting the Manage Database Trace permission to other users for a management database or an audit store database.

Controlling Audit Trail Events

By default, audit trail events are recorded when users log on, open applications, select roles that elevate their privileges, and perform other tasks. You can use domain group policies to control the global location of the audit trail events. For example, you might want to store audit trail events in the audit store database instead of the Windows event Application log if you want to make them available for querying and reports.

You can also override domain group policy and configure local or category-specific audit trail targets using a local administrative template or group policy.

To configure global or per-category audit trail targets using an ADM administrative template:

> **Note**: These settings override the settings defined in the **Set global audit trail targets** group policy.

1. Open the Group Policy Object Editor to display Local Computer Policy, and select **Computer Configuration > Administrative Templates**.
2. Right-click, select **Add/Remove Templates**, then click **Add**.
3. Navigate to the AuditManager folder, select auditrail.adm, click **OK**, then click **Close**.
4. Open the Classic Administrative Templates folder and select **AuditTrail**.
5. Specify global or separate targets for audit trail events:
   - Enable **Set global audit trail target settings** to configure a single location for audit trail events for Access Manager and the Agents.
   - If you want to have separate targets for audit trail events, you can enable the other audit trail group policies to override the global policy setting with a different target.
6. Specify the location for saving audit trail events, and then click **OK**:
   - 0 to disable audit trail events
   - 1 to store audit trail events in the audit store
   - 2 to send audit trail events to the Windows event Application log
   - 3 to send audit trail events to both the audit store and the Application log.

To configure per-category audit trail targets using a local group policy from an XML template:

> **Note**: These settings override the settings defined in the Set global audit trail targets group policy.

1. Ensure that the Audit Trail Settings were updated with the most recent XML template.
2. Open the Group Policy Object Editor to display **Local Computer Policy**, and select **Computer Configuration > Audit Trail Settings**.
3. In **Audit Trail Settings**, separate folders for each audit trail category contain **Send audit trail to Audit database** and **Send audit trail to log file** group policies. Enable these group policies in each category that you want to configure to use a specific audit trail target. The target that you specify for each category is used instead of the target specified in the **Set global audit trail targets** group policy.

Summary of Audit Trail Events

Different components log different audit trail events. For example, the auditing and authorization services on a managed Windows computer track successful logon attempts and the use of Window access rights. Access Manager audit trail events record changes to the configuration of zones, such as the delegation of administrative tasks, the assignment of roles, and changes to the user and group profiles in a zone. For your reference, the following sections summarize the audit trail events recorded by Agents on managed Windows computers.

Additional audit trail events for Access Manager, Audit Analyzer, Audit Manager, and UNIX commands can be recorded in the target you specify for the audit trail. The event message provides detailed information about the operation performed or unsuccessfully attempted, including in most cases the reason the operation was unsuccessfully.

For a complete list of audit trail event identifiers and their corresponding descriptions, see the AuditTrailEvent.xml file provided in the Documentation folder. This file is generated directly from the underlying source code and provides the most up-to-date information about the events on which you can query and report.

Offline MFA Profile Authentication

In some environments, using an offline MFA profile for multi-factor authentication is not compatible with FIPS mode. See the *Multi-factor Authentication Quick Start Guide* for details about this restriction.

Authentication Service Known Issues

When troubleshooting, be aware of the following issues and constraints:

- Import users and groups before importing the sudoers file (Ref: IN-90001).

  Sudoers Import creates user roles but not the users. It is recommended that you import users and groups prior to importing the sudoers file. Otherwise, no sysRights are created for the users.

- Pre-create computers before importing computer role from sudoers file (Ref: IN-90001).

  The computers contained in the sudoers file must either be joined to a zone or pre-created.

- Delegating zone administration permissions for SFU zones (Ref: IN-90001)

Delegate permissions to add, remove or modify users for SFU zone are not supported.

- Users with rights to import user and groups into a zone also gain rights to modify profiles (Ref: IN-90001)

  Any users who are given the right to "Import users and groups to zone" are automatically also given the right to "Modify user/group profiles".

- Using domain local groups to manage resources (Ref: IN-90001)

  Domain local groups can only be used to manage resources in the same domain as the group. So, for instance, a domain local group in domain A may be used to manage a computer in domain A but not one in domain B, despite a trust relationship between the two domains.

- Domain local groups from other domains shown in search dialog (Ref: IN-90001)

  When using the search dialog in the Access Manager to delegate zone control to a group, domain local groups from child domains will be shown incorrectly in the results and should be ignored. The search results when using the ADUC extension do not show these domain local groups.

- Analyze forest and SFU zones (Ref: IN-90001)

  The analyze forest feature in the Access Manager does not report empty zones or duplicated users or groups in a SFU zone.

- Working with users that have more than one UNIX mapping (Ref: IN-90001)

  Authentication Service supports Active Directory users that have more than one UNIX profile in a zone. However, if you are upgrading from DirectControl 4.x or earlier and have existing users with more than one UNIX mapping, you should use DirectControl 5.0.0 or later to remove all but one of the UNIX profiles for each of these Active Directory users and then re-add them.

  In addition, you should always use DirectControl console 5.0.0 or later when modifying these users.

- In the Profile tab of the Properties page of a computer joined to a hierarchical zone, you cannot move this computer to a classic zone. Nor can you move it to a zone in another domain. There are no such limitations with a computer joined to a classic zone. (Ref: IN-90001)

- Extra results when analyzing duplicate service principal names (Ref: IN-90001)

  When running the Analyze / Duplicate Service Principal Names report, kadmin/changepw is incorrectly returned as a duplicate. The SPN is actually found multiple times, but this is by Microsoft design as it is the default account for the Key Distribution Center service in all domains.

- Secondary groups not imported from XML files (Ref: IN-90009)

  Using the Import Wizard to import user information from XML files does not import secondary group membership.

This chapter provides a summary of the command line programs you can run on computers that have the Agent for Windows installed to perform troubleshooting and administrative operations.

Using CopyGroup and CopyGroupNested

The CopyGroup and CopyGroupNested commands help you provision users when there are trust relationships between domains. You can use them to mirror group membership and group hierarchy from a trusted domain and forest to a target domain and forest.

These utilities are located in the Zone Provisioning Agent's **Tools** folder.

To use these command line utilities, you must have an account that can log on to the trusted source domain and the target domain. The account should also have read permission on the source domain and permission to update the target domain.

For example, assume you have configured the AJAX domain to have a one-way trust with the DEVOPS domain and you have your Active Directory users and groups defined in the DEVOPS domain. If you want to allow the users and groups in the DEVOPS domain to log on to computers that are joined to the AJAX domain, you can log on to the AJAX domain controller with an account that has administrative privileges in both the AJAX and DEVOPS domains, then run the CopyGroup utility to mirror the group membership from a group in the DEVOPS source domain as zone users in the AJAX target domain.

For more information about the command line arguments and options for these utilities, see the usage message displayed for each utility.

Using dzinfo

The dzinfo command line program provides detailed information about the effective rights, role definitions, and role assignments for a specified user. The command output includes all of the same information that you can view using the Authorization Center as described in Using the Authorization Center directly on managed computers. However, using dzinfo as a command line utility allows you to view and capture all of the output from the command in a single window, which you can then save as a text file for troubleshooting and analysis or in reports.

The syntax for the dzinfo program is:

dzinfo [/v] [*user_name*] [/h]

The /v is an optional argument that enables you to view verbose output for the command. The *user_name* is an optional argument that enables you to view information for the specified user account. However, you must be logged on as a local administrator to specify the *user_name* argument. If you log on with an account that does not have local administrative privileges you cannot return authorization information for another user account.

If you run the dzinfo command without the *user_name* argument, the command returns authorization information for the currently logged-on user account.

The command returns detailed information about the rights, roles, and role assignments for the specified user (richl in the AJAX domain) similar to the following:

From the Access Manager

Effective roles for AJAX\richl:

Domain Admin/portland

Zone: CN=portland,CN=global,CN=Zones,OU=Acme,DC=ajax,DC=org

Status: Active

Windows Login/global

Zone: CN=global,CN=Zones,OU=Acme,DC=ajax,DC=org

Status: Active

Effective Login Rights for AJAX\richl:

Console Login: Permitted

Audit Level: Audit if possible

Remote Login: Permitted

Audit Level: Audit if possible

PowerShell Remote Access: Permitted

Audit Level: Audit if possible

Role Assignments for AJAX\richl:

Domain Admin/portland

Status: Active

Account: AJAX\richl

Scope: Zone

Zone: ajax.org/Acme/Zones/global/portland

Local Role: No

Network Role: Yes

Effective: Immediate

Expires: Never

Windows Login/global

Status: Active

Account: AJAX\Domain Admins

Scope: Zone

Zone: ajax.org/Acme/Zones/global

Local Role: Yes

Network Role: No

Effective: Immediate

Expires: Never

Role Definitions:

Domain Admin/portland

Status: Active

Description: None

Zone: CN=portland,CN=global,CN=Zones,OU=Acme,DC=ajax,DC=org

Login Permitted: No

Audit Level: Audit if possible

Rescue Right: No

Require MFA: No

Available Hours: All

Rights:

ADUC/portland

Type: Application

Description: None

Priority: 0

Run As: AJAX\Administrator

Application: mmc.exe

Path: C:\Windows\system64

C:\Windows

C:\Program Files

C:\Program Files (x86)

C:\Windows\SysWOW64

Arguments: "C:\Windows\system64\dsa.msc"

Match Case: No

Require Authentication: No

Application Criteria:

None

Domain Admin Network Access/portland

Type: Network Access

Description: None

Priority: 0

Run As: AJAX\Administrator

Require Authentication: No

Windows Login/global

Status: Active

Description: Predefined system role for general Windows login users.

Zone: CN=global,CN=Zones,OU=Acme,DC=ajax,DC=org

Login Permitted: Console & Remote & PowerShell Remote

Audit Level: Audit if possible

Rescue Right: No

Available Hours: All

Rights:

None

Computer is joined to zone ajax.org/Acme/Zones/global/portland

Auditing for AJAX\richl:

Session ID 2:

Desktops:

Default: Not currently auditing.

Auditing is not available on this computer.

Using dzjoin

The dzjoin command line program enables you to automatically join users to the zone in which their roles and rights are assigned, or to join them to a specific zone by zone name, when they log on to their computer. The dzjoin command line program is particularly useful for organizations that use non-persistent virtual desktop infrastructures.

The syntax for the dzjoin command is:

dzjoin [/c <domain controller>] [/d] [/u <username>] [/f] [/h] [/r [y|n|yes|no]] {/z <zonename> | /s | /v]

Note: {/b}If the u option is specified but no password is found in the redirected input, you will be prompted for a password.

/c	Specify a domain controller to connect to.
/d	Retrieve zone data before restarting
/u	Specify the user name to join zone using custom credentials. The user name must be in the format: USER@DOMAIN or DOMAIN\USER. The credentials are for remote access only. For the password, you can specify by redirected input. Otherwise, this tool will prompt user for password.
/f	Suppress any warnings and/or questions.
/h	Displays the command help.
/r	Suppress the restart warning and specify to restart machine, if required, after joining zone. If no restart is required, this option is ignored. If no argument is provided, e.g. '/r', the default is to restart (example: '/r yes').
/z	Join a zone using the zone name. If the zone name is not unique, use the canonical name instead.
/s	Join to the zone where this computer is already pre-created in the zone or had previously been joined to the zone (but remotely left in a disconnected situation).
/v	Display the agent version.

Note: You can also use the PowerShell command Join-CdmZone to join a zone.

Using dzleave

To leave a zone, use the dzleave command. The syntax for the dzleave command is:

dzleave [/c <domain controller>] [/u <username>] [/a|/f] [/r [y|n|yes|no]] [/v] [/h]

/a	Remove the role assignment from the computer zone.

/c	Specify a domain controller to connect to.
/u	Specify the user name to leave zone using custom credentials. The user name must be in the format: USER@DOMAIN or DOMAIN\USER. The credentials are for remote access only. For the password, you can specify by redirected input. Otherwise, this tool will prompt user for password.
/f	Suppress any warning and/or question(s). In case the domain cannot be contacted, this tool will perform a local zone leave automatically.
/h	Displays the command help.
/r	Specify whether to restart machine, if required, after leaving zone without prompt. If no restart is needed, this option is ignored. If no argument is provided, example: '/r', the default is to restart ('/r yes').
/v	Show the agent version.

Note: You can also use the PowerShell command Exit-CdmZone to leave a zone.

Using dzdiag

The dzdiag command line program provides detailed diagnostic information for the local computer. The command output includes all of the same information that you can view by clicking Diagnostics on the Troubleshooting tab as described in Running diagnostics and viewing logs for the agent.

The syntax for the dzdiag command is:

dzdiag [/h] [/o]

The /h is an optional argument that displays the command help.

The /o is an optional argument that allows you to output just the offline MFA provisioning information. You can use this option to see if a user has configured an offline MFA profile or not and details about their offline MFA configuration.

You must be logged on as a local administrator to run the dzdiag command.

The command returns detailed information about desktop sessions similar to the following:

Product: Server Suite *version-number* (*build-number*)
Computer: SERVER01
Joined Domain: acme.local
Zone: acme.local/Program Data/Centrify/Zones/global <!---TODO update path ---> Auditing: Available
Agent State: Connected
Time: 2018-10-04 17:41:41.491 -07:00
Session information:
Session 3
SAM Name: SERVER01\Administrator
Logon Type: Console
Always Audit: Yes
Desktops:
Default
GUID: 3e2c9799-b398-459f-a7a2-ed3a5359af3f
DZ Logon Id: (0x0)
Local Role: Self
Network Roles: Self
Audit Status: Currently Auditing
UAC Restrictions: No
Network Drives: No

Logon information:

Logon ID (0x5bd925)
Logon GUID: 50972030-e9ed-45dc-b7b7-ecf588ef152d
Base Logon ID: (0x1aff6e)
Base SAM Name: ACME\admin
ElevatedAccount: (ElevatedSelfAccount, AdditionalGroups=(count=1, items=(S-1-5-32-544)))
Local Role: Windows Login/CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=local
Network Roles: None
Should Audit: Yes
Logon ID (0x5c2fe6)
Logon GUID: 053ef6cd-10cc-4383-b614-437c1a2067e3
Base Logon ID: (0x1aff6e)
Base SAM Name: ACME\admin
ElevatedAccount: (ElevatedSelfAccount, AdditionalGroups=(count=1, items=(S-1-5-32-544)))
Local Role: Windows Login/CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=local
Network Roles: None
Should Audit: Yes
Logon ID (0x5deca8)
Logon GUID: ce0da851-90f5-4cb6-a71b-25e2b116be75
Base Logon ID: (0x1aff6e)
Base SAM Name: ACME\admin
ElevatedAccount: (ElevatedServiceAccount, ServiceAccount=S-1-5-21-1132289714-2257106472-2904894658-500)
Local Role: Windows Login/CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=local
Network Roles: None
Should Audit: Yes
Logon ID (0x613c40)
Logon GUID: 8ca4e342-4f4a-4e85-8e05-4d1332272c31
Base Logon ID: (0x1aff6e)
Base SAM Name: ACME\admin
ElevatedAccount: (ElevatedServiceAccount, ServiceAccount=S-1-5-21-1132289714-2257106472-2904894658-1108)
Local Role: Windows Login/CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=local
Network Roles: None
Should Audit: Yes

Domain last access information:
Forest acme.local: Connected and Agent can authenticate
Domains:
acme.local (ACME): Connected

The offline MFA provisioning information:
None

Multi-factor Authentication information:
Platform Instance: https://tenant.my.centrify.net/ <!---TODO update path ---> Last Used Platform Instance: <none>
Platform Certificate Exists: No
Disable Web Proxy: No
AD Site: Default-First-Site-Name
Platform Instance Override: <none>
Connector Override: <none>
MFA Enabled (NotJoined): No
Platform Instance (NotJoined): <none>
Web Proxy: <none>

Connectors:
Connector: server01.acme.local
FQDN: server01.acme.local
Tenant: https://tenant.my.centrify.net/ <!---TODO update path ---> Last Known Availability: Yes
Last Access Time: -
IWA Enabled: Yes
IWA HTTPS Port: 8443

Proxy Enabled: Yes
Proxy Server: server01.acme.local:8080
AD Site: Default-First-Site-Name

Using dzrefresh

The dzrefresh command line program enables you to refresh the authorization cache from a Command Prompt window. Running the dzrefresh command provides the same functionality as clicking Refresh on the Troubleshooting tab in the local agent configuration panel as described in Performing cache operations.

The syntax for the dzrefresh command is:

dzrefresh

You must be logged on as a local administrator to run the dzrefresh command. The command output indicates whether the refresh of the authorization cache is successfully initiated.

Using dzflush

The dzflush command line program flushes the authorization cache and reloads all authorization information from Active Directory. Depending on the size of the authorization store, users might experience a temporary loss of the ability to use the rights granted to them while the authorization information is reloaded. To prevent any loss of access privileges, in most cases you should use the dzrefresh command instead of the dzflush command to ensure that the agent is using the latest authorization information. You should only use the dzflush command if Delinea Support recommends that you do so.

The syntax for the dzflush command is:

dzflush [/h] [/l]

/h	Show the command usage.
/l	Synchronize local Windows account information between Access Manager and the Windows systems where local account management is enabled. Note: Local account management is not supported on domain controllers.

You must be logged on as a local administrator to run the dzflush command. The command output indicates whether the authorization cache is successfully flushed.

Using dzdump

The dzdump command line program enables you to view and capture the current content of the authorization cache. You can use command line options to control the information contained in the output for the command.

The syntax for the dzdump command is:

dzdump [/d [*directory-path*]] [/w=*screen-width*] [/s] [/n] [/g] [/l] [/a]
[/r] [/i] [/t] [/z] [/u] [/h]

If you specify no command line arguments, the dzdump command returns complete inmemory information from the authorization agent (dzagent) cache. You can use the following command line arguments to refine the output for the command:

/d	Dump cache files from the default location or a specified location. You can use this option with a directory path to dump cache files from a specified location. For example, to dump cache files from the directory C:\AcmeAZstore:/d=C:\AcmeAZstore Note that you cannot use the /d option to dump cache files directly on a computer where the Agent for Windows is currently running. However, you create a copy of the cache, then dump the cache from the saved copy. For example, copy all files in the cache directory—the default location for cache directory is <!---

	TODO update path ---> c:\ProgramData\Centrify\DirectAuthorize\Cache—to a temporary directory. You can then dump the authorization cache by running dzdump and specifying the temporary location.
/w	Use the specified *screen-width* for word-wrapping the command output. If you don't specify this options, the default screen width is 80 characters. To disable word-wrapping of the command output, specify a *screen-width* of zero. For example: /w=0
/s	Display security identifier (SID) mappings
/n	Display name mappings
/g	Display assignee mappings
/l	Display assignments in the joined zone hierarchy
/a	Display assignments for security identifiers (SID)
/r	Display role definitions
/i	Display right definitions
/t	Display access token information
/z	Display zone hierarchy
/u	Display recent user logon activity
/h	Displays the command help

You can use any combination of display options to display only the information of interest. If you do not specify any display options, the dzdump command displays all of the information in the authorization cache.

You must be logged on as a local administrator to run the dzdump command. You should note that the command output from a dzdump command can contain sensitive information. You should only use the dzdump command if Delinea Support recommends that you do so.

Depending on the display options you specify, the command returns detailed information about the authorization cache.

Using runasrole

The runasrole command-line program enables you to run a specified Windows application using a specified access role. You can use command line options to control whether the role is used as a local role, a network role, or both, and whether to use the current environment or the environment variables associated with the "Run As" user account. The runasrole command line program is equivalent to selecting the Run with Privilege menu option when right-clicking an application shortcut or executable.

The syntax for the runasrole command is:

runasrole /role:*role*[*/zone*] [options] *application* [*argument*]

runasrole /localrole:*role*[*/zone*] [options] *application* [*argument*]

runasrole /networkrole:*role*[*/zone*] [options] *application* [*argument*]

You must specify the role to use in the *rolename*/*zonename* format. You must also specify an appropriate path to the *application* you want to access, including any required or optional arguments.

You can use the following command line arguments and options with the runasrole command:

/role	Use the role name you specify as both a local role and a network role. You can specify this option to run an application locally and access a remote server using the same role, if applicable. You should only use this option if the role you are assigned and want to use has both local and network access rights defined.
/localrole	Use the role name you specify as a local role.
/networkrole	Use the role name you specify as a network role.
/env	Use the current environment variables instead of the environment variables associated with the "Run As" user account.
/netdrives	Use mapped network drives when running an application with the selected role. By default, you cannot use mapped network drives that are associated with you logged-on user account when running applications using a role with elevated privileges. If you want to use a mapped network drive when accessing an application using a selected role, include the /netdrives option in the command line.
/removetimestamp	Remove the grace period on Windows authentication and MFA for the current user session.
/wait	Prevents the runasrole program from exiting immediately after opening the specified application. If you specify this option, the runasrole program starts the specified application and waits until the application session ends before exiting. When the application session ends, the runasrole program exits and returns the same result code as the application. If you specify this option and the application is a command line utility, the runasrole program redirects the application's input and output to the command line console. You should note that some applications use a Microsoft API that does not support redirection of standard input and output. For applications that don't support redirection, the /wait option has no effect and is ignored.
/h	Displays the command help.

Examples

To use the same role to open the Computer Management application locally and access a remote server in zone1, you might run a command similar to the following:

runasrole /role:role1/zone1 mmc.exe c:\windows\system64\compmgmt.msc

To use the role named SQLdba from the finance zone as a local role to open the Services application, you might run a command similar to the following:

runasrole /localrole:SQLdba/finance mmc.exe c:\windows\system64\services.msc

To use role1 from zone1 as a local role to open the Computer Management application and use network access rights from role2 in zone2, you might run a command similar to the following:

runasrole /localrole:role1/zone1 /networkrole:role2/zone2 mmc.exe compmgmt.msc

To open the Services application using the role named SQLdba from the finance zone and have the runasrole program remain open until you close the Services application, you might run a command similar to the following:

runasrole /wait /role:SQLdba/finance mmc.exe c:\windows\system64\services.msc

Running an application from a shortcut

In most cases, you can use the runasrole program to run specified Windows applications using the application shortcut. However, there are many different types of application shortcuts and the RunAsRole program does not support all of them. You can use the RunAsRole program to execute applications with the following recognized shortcut target extensions:

.bat

.cmd

.cpl

.exe

.msc

.msi

.msp

.ps1

.vbs

.wsf

How to determine whether RunAsRole supports an application shortcut

You can determine whether you can use the RunAsRole program to execute an application from the application shortcut by checking the file extension for the target application in the application's shortcut properties dialog box.

To check the file extension for a target application shortcut

1. Select an application shortcut.
2. Right-click the shortcut, then click **Properties** to display the file properties.
3. Click the Shortcut tab and check the target field.

   If the target file extension displayed is a supported file extension, you can use RunAsRole to execute the application from the application shortcut. You should note that a shortcut target field might include both the file name for the application executable and one or more arguments. As long as the application executable has a supported file extension, you can use RunAsRole to execute the application with the specified arguments from the shortcut. For example, if the shortcut target is C:\Windows\System64\control.exe printers, the application executable C:\Windows\System64\control.exe is a supported file extension with printers supplied as an argument. Therefore, you would be able use RunAsRole to run the application from its shortcut.

Using RunAsAlternate

The runasalternate command line program enables you to log in to an application using an alternate account.

For example, system administrators typically have several accounts, a user account for general log-ins and an administrative account to access specific systems and services.

The syntax for the runasalternate command is:

runasalternate [/account:accountname] application [argument] [/h]

You can use the following command line arguments to refine the output for the command:

application	Run an application using the alternate account set in Privileged Access Service.
argument	(optional) Specify an application argument
/account accountname	Specify the alternate account owned by this user for which the application is to be run. This can be useful in cases where a user has more than one alternate account.
/h	Display the command help

If you have only one alternate account defined, you don't need to specify the /account option.

For more information about alternate accounts, see Enabling users to run applications with alternate accounts.

The Agent for Windows can be installed on Windows computers that are configured to run the Server Core operating environment. Server Core is a Windows installation option that provides a low-maintenance server environment with limited functionality.

Most Agent operations are not affected by running on Server Core. However, there are specific features that are not available or not applicable because of the limitations of the Server Core environment itself. For example, the Run with Privilege menu option is not available on Server Core computers because Server Core does not support Windows Explorer and other graphical user interface applications. However, you can use the runasrole command line utility to run specific applications using a specified role.

Similarly, no Delinea notification area applet or desktop rights are available on Server Core computers. However, you can access the Authorization Center, agent configuration panel, and agent command-line utilities from the Server Core command prompt.

The following list summarizes the Agent for Windows features that are not supported on Server Core computers:

- You cannot create, select, or switch desktops or use any desktop-related features because the Windows desktop is not available on Server Core.
- You cannot select Run with Privilege as a right-click menu option for applications because Windows Explorer is not available on Server Core.
- You cannot open the Authorization Center or access the Delinea notification area applet because the Windows desktop and Windows Explorer are not available on Server Core.
- You cannot open applications such as the agent configuration panel from Start menu shortcuts because the Windows desktop and Windows Explorer are not available on Server Core.

You should note that only the Agent for Windows is supported for the Server Core environment. A small number of other Server Suite components for Windows support a command line interface, but are not configured to support a Server Core environment.

Server Core Supported PLatforms

Delinea supports the following versions of the Server Core environments:

- Windows Server 2008 R2 Server Core
- Windows Server 2012 Server Core
- Windows Server 2012 Minimal Server Interface
- Windows Server 2012 R2 Server Core
- Windows Server 2012 R2 Minimal Server Interface

You should note that Server Core is not supported on Windows Server 2008 because Windows Server 2008 Server Core does not support any version of the .NET Framework. The Agent for Windows requires the .NET Framework. For more information about the supported libraries and .NET functionality on Server Core, see the reference material available on the Microsoft Developer Network website for the operating system you have deployed.

For general information about Server Core on Windows Server 2008 R2, see: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc753802(v=ws.10)

For general information about Server Core on Windows Server 2012 R2, see: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831786(v=ws.11)

Installing the Agent on a Computer Running Server Core

You cannot use the autorun.exe or the setup.exe program to install components on a computer that is configured to run as a Server Core environment. Instead, you must install from Microsoft Installer (.msi) files using the msiexec command-line program.

To install the Agent for Windows on Server Core:

1. Use the Deployment Image Servicing and Management (DISM) or another command-line tool to enable the .NET Framework.

   For example, if you are using Windows Server 2012 or later and the .NET Framework is located on the installation media in the D:\sources\sxs folder, use the following command:

   DISM /Online /Enable-Feature /FeatureName:NetFx3 /All /LimitAccess /Source:D:\sources\sxs

   To install .NET Framework on Windows Server 2008 R2, run the following commands to enable the required features:

   Dism /Online /Enable-Feature /FeatureName:NetFx2-ServerCore-WOW64

Dism /Online /Enable-Feature /FeatureName:NetFx3-ServerCore-WOW64

Dism /Online /Enable-Feature /FeatureName:NetFx2-ServerCore

Dism /Online /Enable-Feature /FeatureName:NetFx3-ServerCore

2. Copy the Agent for Windows files to the Server Core computer.

   For example:

   copy D:\Common\Centrify* C:\Agent

   copy D:\Agent* C:\Agent

3. Install the Common Component service using the .msi file.

   For example, to install the Common Component on a computer with 64-bit architecture, you might use the following command:

   msiexec /i "Common Component64.msi" /qn

4. Install the Agent for Windows using the .msi file.

   Run the following command:

   msiexec /i "Agent for Windows64.msi" /qn

5. Restart the computer with the appropriate shutdown options to complete the installation and start agent services.

   For example, you might run the following command:

   shutdown /r

   Note that restarting the computer is not required if you install only auditing features.

Opening Consoles on Server Core Computers

Because the primary interface for the Server Core environment is a command prompt with only limited support for graphical user interface features, you must use the command line to open the consoles that enable you to join or leave a zone, view your rights and roles, and configure agent settings.

Joining a Zone

One of the first tasks after installing the Agent for Windows is to join a zone. You can do by launching the agent configuration panel from the command prompt.

To open the agent configuration panel to join a zone:

1. Navigate to the Agent for Windows installation directory.
2. By default, the agent files are installed in the C:\Program Files\Centrify\Agent for Windows directory. <!---TODO update path --->
3. Run Centrify.DirectAuthorize.Agent.Config.exe. <!---TODO update filename--->
4. Click **Change**.
5. Click **Browse**.
6. Type all or part of the zone name, click Find Now, then select the zone to join and click **OK**.
7. Click **Close** to exit the agent configuration panel.

If you later need to change the zone, run diagnostics, refresh the authorization cache, or view or modify log settings, you can run <!---TODO update filename---> Centrify.DirectAuthorize.Agent.Config.exe to perform those tasks.

Viewing Authorization Details

By default, identity management, privilege management, and audit and monitoring service features are enabled after you install and configure the Agent for Windows. To see details about your rights, role definitions, role assignments, and auditing status, you can launch the Authorization Center from the command prompt.

To open the Authorization Center on a computer with the Server Core operating system:

1. Navigate to the Agent for Windows installation directory.

   By default, the agent files are installed in C:\Program Files\Centrify\Agent for Windows directory. <!---TODO update path--->

2. Run Centrify.DirectAuthorize.Auth.Center.exe. <!---TODO update filename--->

Configuring Auditing Options

By default, identity management, privilege management, and audit and monitoring service features are enabled when you install the Agent for Windows. To configure audit and monitoring service options and specify the audit installation for the agent, you can launch the agent configuration panel from the command prompt.

To open the agent configuration panel to configure auditing features:

1. Navigate to the Agent installation directory.

   By default, the agent files are installed in the <!---TODO update path---> C:\Program Files\Centrify\Audit\Agent directory.

2. Run agent.configure.exe.
3. Click **Configure**.
4. Select a color quality, then click **Next**.

   Because the Server Core operating system uses very few graphical elements, in most cases you should accept the default setting of Low for the color quality. This setting minimizes the storage requirements for auditing if you have enabled video capture auditing.

5. Accept the default offline data location and maximum size or type a different location, then click **Next**.

   You can also drag the slider to change the maximum percentage of the drive the offline data can consume. In most cases, however, you should leave the default setting unchanged.

6. Select the audit installation, then click **Next**.
7. Review your configuration settings, then click **Next**.
8. Click **Finish** to close the configuration wizard.
9. Click **Close** to exit the agent configuration panel.

Running Command Line Programs

The Agent for Windows includes several command line programs for performing administrative tasks. The following command line programs are supported on Server Core computers:

- dzinfo
- dzjoin
- dzdiag
- dzrefresh
- dzflush
- dzdump
- runasrole

For more information about the command line options or output for these commands, see Using Windows command line programs or run the command with the /help option.

Unsupported Windows Server 2012 Features

Windows Server 2012 includes support for claims, compound authentication, and Kerberos armoring. The core Agent for Windows does not provide support for these advanced authentication features. To take full advantage of these advanced authentication services, however, requires you to make the following changes to your environment:

- Deploy Dynamic Access Control.

- Upgrade all of your domain controllers and application servers to Windows Server 2012 or later.
- Upgrade all of your workstations to Windows 8 or later.
- Raise the domain functional level to Windows Server 2012.

If you have a mixed environment that includes Windows 7 and Windows 8 or later workstations and Windows Server 2008 or Windows Server 2008 R2 domain controllers, you can configure the administrative template for claims, compound authentication, and Kerberos armoring to use the Not supported option (default).

To use the Supported configuration option, you must deploy Dynamic Access Control, configure Windows 8 and later client-side support for claims, compound authentication and Kerberos armoring, and ensure you have domain controllers running Windows Server 2012 to handle the authentication requests for those computers. You should not install the Agent for Windows on any computers configured to support claims, compound authentication and Kerberos armoring to prevent authentication failures.

In addition, Server Suite does not provide any specific support for authenticating access to Server Message Block 3.0 (SMB3.0) file shares that are supported in Windows Server 2012. The SMB protocol operates as an application layer for providing shared access to computers, printers, and other devices. This protocol has been extended to provide shared access to virtual machines and SQL user databases.

This *Quick Start Guide* provides a brief summary of the steps for installing and getting started with Server Suite software. For more information about any step, see the appropriate sections in the Unexpected Link Text or Unexpected Link Text.

1. Run the setup program for Authentication & Privilege components on a Windows administrator's workstation.

   The setup program simply copies the necessary files to the local Windows computer, so there are no special permissions required to run the setup program other than permission to install files. Follow the prompts displayed to select which components to install.

2. Open Access Manager to start the Setup Wizard and create an organizational structure and the containers for Licenses and Zones.

   In the Setup Wizard, you can accept the default organizational structure or create a custom organizational unit for Server Suite objects, add license keys, and configure a few basic permissions and setup options.

3. In Access Manager, create a new zone with the default options. For example, create a new zone named **Demo**.

4. In Access Manager, add Active Directory users to the new zone.

   - Select the new **Demo** zone.
   - Right-click, select **Add User** to Select User Type, then select Active Directory users to search for and select existing Active Directory users.
   - Select **Define user UNIX profile** and deselect assign roles.
   - Accept the defaults for all fields.

5. Create a child zone.

   - Select the **Demo** zone.
   - Right-click, then select **Create Child Zone**.
   - Type a name for the zone, for example, **Child1** and an optional description, then click **Next** and **Finish** to create the new child zone.

6. Assign a role for the users you added to the Demo zone.

   User profiles are inherited by child zones, so the users you added to the Demo zone automatically have a profile in Child1. To log on to a computer, users must have a profile and a role assignment. You can assign the default *UNIX Login* role to enable users to log on.

   - Expand **Child Zones**, **Child1**, and **Authorization**.
   - Select **Role Assignments**, right-click, then click **Assign Role**.
   - Select the **UNIX Login** role from the results and click **OK**.
   - Click **Add AD Account**.
   - Search for and select one of the Active Directory users you added to the Demo zone, then click **OK**.

7. On the Linux or UNIX computer, log on as root. if you are installing on a computer running Linux or UNIX.

8. Run the install.sh command.

   ```
   ./install.sh
   ```

   The installation script checks whether the computer meets all system requirements, such as a supported operating system, available disk space, DNS and network connectivity, and your Active Directory configuration.

   If the computer meets all requirements, you can choose to install all Server Suite, or a customized set of services. You can also choose whether to automatically join the domain and restart the local computer to complete the installation. After you make your selections, the script installs a platform-specific Server Suite Agent and any other packages.

   Alternatively, you can install using a native package manager or another software distribution utility. The command line syntax and the agent package name will depend on the operating system on which you are installing.

   To manually join the domain after installation, use the adjoin command. In either case, you must specify the zone to join. For example, if you created the Child1 zone, you might run a command similar to this:

   ```
   adjoin myDomain -z Child1
   ```

   In Step 4, you created a profile for an Active Directory user in the Demo zone. In Step 6, you assigned the user the UNIX Login role. You can now verify authentication by logging off as root and logging on to the computer you just joined to the Active Directory domain with the Active Directory user

account and password you assigned the UNIX Login role.

That's it!

From here, if you want to explore further, you can:

- Create and assign additional roles to users
- Create new child zones
- Import existing UNIX users and groups
- Override user attributes in child zones
- Set group policies for UNIX computers and users
- Run reports
- Import and manage NIS maps in Active Directory

For more information about any topic, see the Server Suite documentation set.

This guide is intended for UNIX or Windows administrators who intend to configure multi-factor authentication for computers managed by Server Suite.

Configuration information for Delinea customers who are not using Server Suite to manage their environment, but want to configure multi-factor authentication to log in Windows computers, should go to Downloading the Server Suite Agent for Windows.

There are two separate scenarios for which you might want to require multi-factor authentication:

- **Login** access to Server Suite-managed computers.
- As part of a **re-authentication** process so that users who are attempting to use Application, Network, and Desktop rights on Windows machines, or command rights with elevated privileges or in a restricted shell on UNIX machines, must provide a password and another form of authentication before they can execute the selected command.

With these two scenarios in mind, you can configure multi-factor authentication based on user roles or computer roles, for specific applications, or for individual commands. You can also skip multi-factor authentication for applications that do not support it or for other reasons on a case-by-case basis by enabling and applying group policy or by setting configuration parameters.

You can configure multi-factor authentication for users logging on to Server Suite-managed computers to improve the security of physical or virtual data centers. You can do this by assigning the predefined require MFA for login role to users who are required to provide more than one form of authentication. Alternatively, for UNIX and Linux roles, you can also create custom role definitions with the **Require multi-factor authentication for login** system right selected. Because the Windows Login role can be assigned to local accounts, there is no system right for multi-factor authentication, therefore you must assign users the require MFA for login role.

Roles and role assignments are important when configuring multi-factor authentication for login access to Server Suite-managed computers in hierarchical zones.

Before configuring multi-factor authentication, you should be aware that multi-factor authentication for Server Suite-managed computers relies on the infrastructure provided by the Delinea Platform.

Note: For Linux and UNIX computers, logging on requires a PAM application such as login, ssh, or a desktop manager. Most programs that enable users to log in support multi-factor authentication. However, some desktop manager programs that run on older operating systems might not support multi-factor authentication.

Multi-Factor Authentication and Smart Card PIN Login

A smart card user logging in by way of a Personal Identification Number (PIN) will not authenticated by multi-factor authentication. (Ref: CS-38641)

If you have installed Server Suite, you can require multi-factor authentication when users perform operations with an elevated access right, in addition to requiring multi-factor authentication when users log in. For Linux and UNIX computers, for example, you can also create command rights that require multi-factor authentication when executing commands using elevated privileges (dzdo) or in restricted shell (dzsh) environments. For Windows computers, you can create desktop, application, and network access rights that require two-step authentication to use the elevated privileges associated with the desktop, application, or network access.

Before configuring multi-factor authentication for any type of access right, you need to perform some preliminary steps to prepare your environment.

Because multi-factor authentication for Server Suite-managed computers relies on the infrastructure provided by the Delinea Platform, there are steps that require access to a Delinea Platform instance and the administrative portal. As a preview, here are the steps involved in preparing the identity platform to support multi-factor authentication for Server Suite-managed computers:

- Register for the **Delinea Platform**.
- Install and configure at least one Cloud Connector for communication with the platform. Your machine account must have login access to the connector machine.
- Verify the users who are required to provide more than one form of authentication have valid **Active Directory accounts** that are active in the platform.
- Add or select the **authentication profiles** that specify the types of authentication challenges to support.
- Create a role with the appropriate **computer members and administrative rights** for multi-factor authentication.
- Verify the **server authentication instance URL** you want to use if you have access to more than one authentication instance.

After you have completed the preliminary steps, you can assign users the predefined require MFA for login role or, for users of UNIX and Linux machines, a custom role with the **Require multi-factor authentication for login** system right to require two-step authentication when logging on. These preliminary steps are also required if you want to create command rights that require two-step authentication when executing commands using elevated privileges (dzdo) or in restricted shell (dzsh) environments on UNIX and Linux machines, or when creating roles with elevated Windows rights.

Multi-factor authentication for Server Suite-managed computers relies on the infrastructure provided by Privileged Access Service and authentication and privilege elevation. Privileged Access Service enables you to securely manage users, roles, policies, devices, and applications in the identity platform. You can also define the types of authentication challenges you support and where the multi-factor authentication rules apply.

Sign Up and Activate Your Account

To get started, you should register for an account in Privileged Access Service if you are not already a subscriber. You can request a free trial or subscribe to Privileged Access Service through the Delinea website.

If you don't already have a subscription, you can start by requesting access to Privileged Access Service by visiting the Delinea website.

After you register for a Delinea account with a valid email address, you will receive an "Activate Your Delinea Account" email followed by a "Your Delinea Account Is Ready Next Steps" email with your account details. Your account details include the user name and temporary password for an administrative account that is a member of the predefined System Administrator role and a unique customer identifier. For example, your email message might have account details similar to the following:

	https://abcd1234.my.centrify.net/manage
Your User Name:	admin_maya.garcia@acme.net
Your Temporary Password:	1234abcepassword (You'll be asked to change this when you log in)
Customer ID:	ABCD1234

Use your account details to log in and set a new password for your administrative account.

Start or Skip the Wizard

After you log in successfully, you will see a Welcome to Privileged Access Service message with the option to start or skip the quick start wizard.

If you click **Start the Wizard**, you are prompted to manage mobile devices, add web applications, add mobile applications, add users, and invite users. You can click Next to skip any or all steps. None of the steps in the wizard are required to set up multi-factor authentication.

If you are only interested in preparing for multi-factor authentication, you can select the **Don't show this to me again** option, then click **Skip**. If you click Skip now, you can run the wizard at any time after configuring multi-factor authentication by clicking **Start Wizard** on the Getting Started dashboard.

If you have not completed these preliminary steps, stop here and verify that you have received the "Your Delinea Account Is Ready - Next Steps" email and that you can log in to the Privileged Access Service platform with the account information in the email.

Plan Multi-Factor Authentication for Server Suite-Managed Computers

Privileged Access Service is most often used to store information about people and devices, to identify different classes of users and devices, and to define the policies that specify what different classes of users and devices can do. To support multi-factor authentication, however, you must also add Delinea-managed computers to the access service.

Any computer that will require multi-factor authentication must also be added as a member of an identity platform-based role. This step is similar to adding computers to a zone. For multi-factor authentication, an identity platform-based role has computers as members and is managed through Privileged Access Service. It is separate from the role definitions and role assignments you manage using Access Manager or other Server Suite components.

The connector is a multipurpose service that enables secure communication between your internal network and Privileged Access Service. Multi-factor authentication requires at least one connector to be installed on your network inside of the firewall. The connector provides the link between your internal Active Directory forest and the Privileged Access Service platform.

You can install more than one connector for your organization to support fail-over and load balancing. You might also want to install more than one connector if you are using multiple instances of Privileged Access Service or have access to more than one customer-specific Identity platform instance URLs. In most cases, you should install at least two connectors in a production environment.

Note: Make sure that the account you're logged into on the computer has the appropriate permissions to install the Cloud Connector.

For details about installing and configuring the connector, please see the following topics in the Privileged Access Service help:

- Installing a connector
- Configuring a connector
- Installing a connector
- Configuring a connector

Establishing a Connector Identity for Multi-Factor Authentication

In order to enable multi-factor authentication for Delinea-managed UNIX and Linux machines, the connector must validate the machine credentials using the Integrated Windows Authentication (IWA) service. To use the IWA service, your connectors must be configured to use an HTTPS-enabled port.

To configure connectors to use an HTTPS-enabled port, you must either download a host certificate issued by Delinea, or upload a host certificate issued by a Certificate Authority (CA) already trusted by your environment.

To configure Windows computers for multi-factor authentication, you must establish an initial trust relationship between the Windows machine and the Cloud Connector. Since the connector accesses the IWA service through a secure HTTPS channel, you must validate the correct certificate during installation when enabling multi-factor authentication for login.

If you are operating in an evaluation environment, and cannot easily set up the required certificate trust relationship, you have the option to skip this step during installation and trust your own connector without enrolling in the IWA service. In this case, the computer is connected directly to the Delinea Platform, and multi-factor authentication can be enabled. Note, however, that this should only be done in an evaluation environment, as it has serious security implications in a live production environment.

If you have chosen not to establish the trust relationship, but wish to do so in the future, you can either leave and then rejoin a zone if you are joined to one, or you can disable and then re-enable multi-factor authentication for login to launch the configuration wizard.

To configure a connector to use a Delinea-issued root certificate

1. In the Admin Portal, click **Settings > Network**.
2. Select the connector you want to configure, and choose **Modify** from the **Actions** menu.
3. In **IWA Service**, click **Download your IWA root CA Certificate** to retrieve the public certificate for the tenant-specific CA certificate issued by Delinea.
4. Click **Download** to download the host certificate issued by Delinea for your connector.

You can export the IwaTrustedRoot.cer trusted root CA certificate issued by Delinea and manually install it on a local computer, or use group policy to distribute the certificate file as a trusted root certificate to multiple computers

Note: Express users cannot use group policies to distribute certificates in bulk to UNIX and Linux computers. To distribute the certificates, you must download and install the certificate in the appropriate directory on each computer.

To Import the Certificate Manually to a Local Windows Computer

1. Right click on the certificate you downloaded in To configure a connector to use a Delinea-issued root certificate.
2. Select **Install Certificate** to start the Certificate Import Wizard.
3. Select **Local Machine** and click **Next**.
4. Select **Place all certificates in the following store** and click **Browse**.

5. Select **Trusted Root Certification Authorities** and click **OK**.
6. Click **Next** and then **Finish** to complete the Wizard.

   A Windows Security Warning may be displayed. Click Yes to finish installing the certificate.

To Export the Certificate for Bulk Group Policy Distribution

1. Select the trusted root certificate you downloaded, right-click, then click **Open**.
2. Click the **Details** tab and click **Copy to file** to start the Certificate Export Wizard, then click **Next**.
3. Select **DER encoded binary X.509 (.CER)** as the file format, then click **Next**.
4. Click **Browse** to select a location on the local server, type a file name and click **Save**, then click **Next**.
5. Click **Finish**.

To Distribute the Certificate using Group Policy

1. Open Group Policy Management to select the group policy object that defines the IP Security policies, then click **Edit**.
2. Click **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Trusted Root Certification Authorities**.
3. Select **Trusted Root Certification Authorities,** right click, and select **Import** to open the Certificate Import Wizard.
4. Click **Next** on the **Welcome** screen.
5. Browse to find the root certificate you downloaded, then click to accept the default values on each screen.
6. **Click Finish** to complete the wizard.

   The root certificate is now in the Active Directory Trusted Root Certification Authorities container. Group policy publishes all certificates in this container to computers joined to the domain. You can also run the gpupdate command from a command prompt to push the certificates to the computers in the domain.

Using a certificate not issued by Delinea with the Cloud Connector

If you want to use a certificate issued by a CA that is trusted by your organization, you must upload the certificate from the Cloud Connector Configuration program. Doing this ensures that the computer credentials can be validated for secure communication between the connector and the authentication server. The issuer of the certificate must also be trusted by resources running agents.

To use an externally issued certificate for a Cloud Connector

1. In the Admin Portal, click **Settings > Network > Centrify Connectors**.
2. Select the connector you want to configure, and choose **Modify** from the Actions menu.
3. Click **IWA Service**.
4. Click **Upload** and navigate to the location of the certificate trusted by your organization.

   Note: You may get the following error while enrolling a Windows agent/machine into the cloud-based CIS with debugging enabled:

```
An error occurred while sending the request. ---\> System.Net.WebException:
The underlying connection was closed: Could not establish trust relationship for
the SSL/TLS secure channel. ---\> System.Security.Authentication.AuthenticationException:
The remote certificate is invalid according to the validation procedure.
```

If so, check your local machine trusted root CA. The server may not have the corresponding DigiCert Global Root CA installed. If so, export the local certificate. Then import the certificate to the server. After that, you should be able to enroll the server.

Verify Open Ports

Multi-factor authentication requires the following ports to be open for inbound communication and domain traffic:

- Port 8080 for HTTP API proxy
- Port 8443 for secure HTTP (HTTPS) connections

Installing the connector automatically sets Windows firewall rules to open these ports. However, if you are using a third-party firewall instead of the default Windows firewall, you should manually modify the port rules to allow the Server Suite Agent for Windows to communicate with the Cloud Connector. Both ports are required because integrated Windows authentication over HTTPS uses port 8443 to enable the connector to receive inbound connections from the agents.

After you have installed and configured at least one connector, you can use either the Admin Portal or your default browser to log in to the Delinea Platform.

To log in and verify settings

1. Open a browser and log in to your customer-specific identity platform instance URL.
2. Type the user name from your account details and the password you set when you activated the service.

   If you see the Welcome message, select the "Don't show this to me again" option, then click **Close**.

   By default, the Getting Started dashboard is displayed in the Admin Portal. The Getting Started dashboard has links to topics that explain important tasks—such as creating roles and adding users—for the Delinea Platform. You will perform similar steps to prepare for multi-factor authentication. However, you can skip those steps for now. For multi-factor authentication, you should first verify some settings on your connector and, if necessary, prepare a new Active Directory group for the computers where you plan to use multi-factor authentication.
3. Click **Settings > Network > Centrify Connectors**.
4. Select the connector and then select **Modify** from the Actions menu to display the connector Configuration.
5. Verify the following options are selected under IWA Service:
   - Enable Web Server

     This option is required to enable integrated Windows authentication for agents and multi-factor authentication.
6. Click **Save**.

After verifying connector settings, you can use Active Directory Users and Computers or other tools to prepare an Active Directory group for the computers where you plan to require multi-factor authentication. Although you can use any existing Active Directory group for this purpose, the steps in this guide assume you will use a new group specifically for multi-factor authentication.

Multi-factor authentication requires computers to be members of an **identity platform role** assigned a specific **administrative right** in the Delinea Platform. You can add individual computers independently without using an Active Directory group. However, using an Active Directory group is the recommended approach and facilitates the deployment of computer roles that link user role assignments to computer groups.

To Add an Active Directory Group for Multi-Factor Authentication

1. Open Active Directory Users and Computers.
2. Select a location, right-click, then select **New** > **Group**.

   For example, if you are using the default deployment structure, you might expand the Centrify organization unit and select Computers, then right-click to create a new group in that organizational unit.
3. Type a group name, select the group scope, and verify the group type, then click **OK**.

   For example, type MFA-Group, select Global for the group scope, and verify the group type is Security, then click **OK**.

Offline MFA mode is not triggered during logon to the computer once the agent has successfully connected to the cloud. Logon will fail if the cloud fails to authenticate the user, the user is not allowed to perform the MFA logon, or if the user is not assigned to any profile in the portal.

To add a user or group to a role in the Admin Portal

1. Create a new role or double-click an existing role that is policy-specified.
2. Click **Members** › **Add**.
3. Enter a search string to locate the Active Directory groups or users you are using that require multi-factor.
4. Select the group or user and then click **Add**.
5. Click **Save**.

After you have prepared an Active Directory group for the computers where you plan to require multi-factor authentication, you can use the Admin Portal to prepare a role for those computers.

To prepare a role in the Admin Portal:

1. In the Admin Portal, click **Core Services**, then click **Roles**.
2. Click **Add Role**.
3. Type a role name and, optionally, a role description.

   For example, type MFA-LinuxComputers as the role name and Role for multi-factor authentication of Linux Computers as the role description, then click **Save**.
4. After naming the role, click **Members**, then click **Add**.
5. Type a search string to locate the Active Directory group you are using for computers that require multi-factor authentication.

   For example, if you created a group called Audited Servers in Preparing a group for Delinea-managed computers, you might type “aud” as the search string to locate that group. Alternatively, you can search for and add individual computers to the role if you are not using an Active Directory group. Adding individual computers to the role, however, is not a scalable approach for most organizations.

   This step creates the link between the Delinea-managed computers and the identity service. There is no change to how you manage the computers you add to the identity service. This link is required to allow Privileged Access Service to provide authentication profiles to managed computers.
6. Select the group, then click **Add**.
7. Click **Administrative Rights**, then click **Add**.
8. Select the **Computer Login and Privilege Elevation** administrative right, then click **Add**.

   This administrative right is only applicable for the computers that are members of the identity platform role. The right does not apply to users and is ignored for any users added as members of the role. In general, you should not add users to any role that is intended for multi-factor authentication on Delinea-managed computers.
9. Click **Save**.

With Server Suite, you can require multi-factor authentication for two distinct situations:

- As part of the **login** process so that users who are attempting to log in to Delinea-managed computers must provide multiple forms of authentication before they are granted access.
- As part of a **re-authentication** process so that users who are attempting to use Application, Network, and Desktop rights on Windows machines, or command rights with elevated privileges or in a restricted shell on UNIX machines, must provide a password and another form of authentication before they can execute the selected command.

To configure the types of authentication challenges allowed in each situation, you can prepare one or more **authentication profiles** in the Admin Portal. If you have already configured authentication profiles for other purposes, you can reuse those profiles for multi-factor authentication or add new profiles specifically for the computers you manage using Server Suite. You can prepare one profile to use for both login access and for the use elevated privileges or you can prepare separate profiles for each situation.

Create an Authentication Profile

The first step in preparing authentication profiles is to create the profile.

To create an authentication profile:

1. Open a browser and log on to Privileged Access Service using your customer-specific URL.
2. Navigate to **Settings** › **Authentication**.

   Three default authentication profiles are available out-of-the-box:

   - **Default New Device Login Profile**: Uses Password for the first challenge and Mobile Authenticator, Text message (SMS) confirmation code, Email confirmation code, or OATH OTP Client for the second challenge with a 12 hours pass-through duration.
   - **Default Other Login Profile**: Uses Password for the first challenge and no secondary challenge with a 12 hours pass-through duration.
   - **Default Password Reset Profile**: Gives the option for users to use Mobile Authenticator, Text message (SMS) confirmation code, Email confirmation code, or OATH OTP Client for the first challenge with a 12 hours pass-through duration.

3. Select an existing Authentication Profile or click **Add Profile**.

   The fields needed to **add new profile**.

   1. Type the authentication profile name.
   2. Select the types of authentication to present for the first challenge.

      Note: The second authentication is not needed. Challenge two is a third mechanism.

   3. Click **OK**.

The pass-through option does not apply to Windows, Linux, or UNIX MFA logins unless you specify otherwise in the policy settings.

Select the authentication mechanism(s) you require and want to make available to users. Some authentication mechanisms require additional configurations before users can authenticate using those mechanisms. See Authentication mechanisms for information about each authentication mechanism.

For example, you can require that the first challenge be the user's account password. Then for the second challenge, users can choose between an email confirmation code, security question, or text message confirmation code.

If you have multiple challenges, Privileged Access Service waits until users enter all challenges before giving the authentication response (pass or fail). For example, if users enter the wrong password for the first challenge, we will not send the authentication failure message until after users respond to the second challenge.

If users fail their first challenge and the second challenge is SMS, email, or phone call, the default configuration is that Privileged Access Service will not send the SMS/email or trigger the phone call. Contact support to change this configuration.

Assign Login Authentication Profiles

The next step is to assign login authentication profiles to policies. In this task, you set up a policy so that if specified conditions are met, the affected users proceed according to a specified authentication profile. If those conditions aren't met, you can specify a default authentication profile or block access entirely.

For example, you could set a policy that says that during work hours of Monday to Friday, 8:00 am to 5:00pm, users log in using an authentication profile that requires a password and a security question. For users logging in outside of those days or times, users will have to login with a password, security question, and an email confirmation code.

As a reminder, you use **authentication profiles** to define the necessary authentication methods to use. You define **authentication rules** to specify where to enforce those authentication profiles inside of a policy set.

To assign a login authentication profile to a policy set:

1. In the Admin Portal, go to **Access** > **Policies** and either click **Add Policy Set** to create a new policy or click an existing policy to edit.
2. Create or edit the policy set and assign it to the desired users or resources.

   For details, see "Creating policy sets and policy assignments" in the Privileged Access Service help.
3. In the Policy Settings area, navigate to Authentication > Centrify Server Suite Agents > and click one of the following settings:

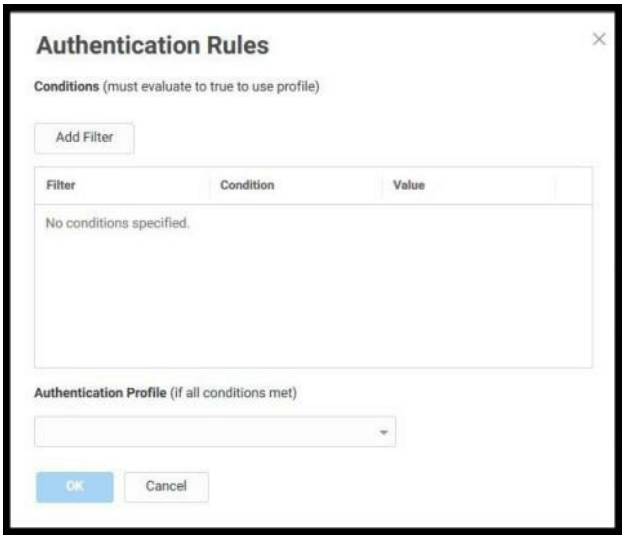
Linux, UNIX and Windows Servers	For Linux and UNIX Servers or Workstations where the Server Suite Agent for *NIX is installed and enabled. For Windows Servers where the Server Suite Agent for Windows is installed and enabled
Windows Workstations	For Delinea-managed workstations where the Server Suite Agent for Windows is installed and enabled. The operating system variant determines if it's a workstation.
Privilege Elevation	For systems where either the Server Suite Agent for *NIX or Serer Suite Agent for Windows is installed and enabled.

Note: For any of the above policy settings, the role assignment associated with this policy must include computer objects or groups in Active Directory and also the "Computer Login and Privilege Elevation" administrative rights.

1. Select **Yes** in the **Enable authentication policy controls** drop-down.

   The Authentication Rules section displays. You use this section to define which authentication profiles apply under which conditions.
2. (Optional) If you want to specify conditions for which different authentication rules apply, click **Add Rule**. Otherwise, proceed to step

   The Authentication Rules window displays.



3. Click **Add Filter**, and then click the same drop-down to spicy which kind of condition.

   For example, you can create a rule that requires a specific authentication method when users access Privileged Access Service from an IP address that is outside of your corporate IP range. Supported filters are:

IP Address	The authentication factor applies as follows: For Privileged Access Service on-premise, the authentication factor is the connector's IP address when you log in. When using HTTP proxy, the authentication factor is the HTTP Proxy server's IP address when you log in. For Privileged Access Service, the authentication factor is the tenant connectors' public IP address when you log in. When using HTTP proxy, the authentication factor is the HTTP proxy server's public IP address when you log in. This option requires that you have configured the IP address range under **Settings** > **Network** > **Corporate IP Range**. **Note**: For Windows machines that can access the Internet, the authentication factor is the machine's IP address when you log in.
Identity Cookie	The authentication factor is the cookie that is embedded in the current browser by the directory service after the user has successfully logged in.
Day of Week	The authentication factor is the specific days of the week (Sunday through Saturday) when the user logs in.
Date	The authentication factor is a date before or after which the user logs in that triggers the specified authentication requirement.
Date Range	The authentication factor is a specific date range.
Time Range	The authentication factor is a specific time range in hours and minutes.
Risk Level	Risk Level: The authentication factor is the risk level of the user logging on to Admin Portal. For example, a user attempting to log in to Privileged Access Service from an unfamiliar location can be prompted to enter a password and text message (SMS) confirmation code because the external firewall condition correlates with a medium risk level. This Risk Level filter, requires additional licenses. If you do not see this filter, contact Delinea support. The supported risk level are: Non Detected — No abnormal activities are detected. Low — Some aspects of the requested identity activity are abnormal. Remediation action or simple warning notification can be raised depending on the policy setup. Medium — Many aspects of the requested identity activity are abnormal. Remediation action or simple warning notification can be raised depending on the policy setup. High — Strong indicators that the requested identity activity is anomaly and the user's identity has been compromised. Immediate remediation action, such as MFA, should be enforced. Unknown — Not enough user behavior activities (frequency of system use by the user and length of time user has been in the system) have been collected.
	For the Day/Date/Time related conditions, you can choose between the user's local time and Universal Time Coordinated (UTC) time.

4. Click the **Add** button associated with the filter and condition.
5. Select the authentication profile you want applied if all filters/conditions are met in the **Authentication Profile** drop-down.

   The authentication profile defines which authentication methods to use. If you have not created the necessary authentication profile, select the **Add New Profile** option in the list (it's at the bottom of the list).
6. Click **OK** to close the Authentication Rules dialog box.
7. If desired, continue adding authentication rules. You can drag the rules to change the order of priority. The highest priority rule is at the top of the list.
8. Select a default profile to be applied if a user does not match any of the configured conditions in the **Default Profile (used if no conditions matched)** drop-down.

   Note: If you have no authentication rules configured and you select **Not Allowed** in the **Default Profile** dropdown, users will not be able to log in to the service.
9. If this policy setting is for Linux, UNIX, and Windows Servers, you have the option to configure how the pass-through duration applies. The pass-through duration is how long before the user needs to re-authenticate, and you define the pass-through duration in the authentication profile (for example, the default is 30 minutes). Select one of the following options:

- **Never (default):** Always prompt for MFA and ignore the pass-through setting.
- **If Same Source and Target**: Apply the pass-through duration if the user is logging in from the same system and where they're logging in to is the same system as compared to the initial login.
- **If Same Source**: Apply the pass-through duration if the user is logging in from the same system as compared to the initial login.
- **If Same Target**: Apply the pass-through duration if the user is logging in to is the same system as compared to the initial login.

10. If desired, you can add multiple policy settings to the same policy set.
11. Click **Save**.

Assigning Privilege Elevation (Re-authentication) Profile

Finally, you must assign privilege elevation profiles.

1. For Elevated Privileges Profile, click **Privilege Elevation Policies > Privilege Elevation**, select **Yes** for Enable authentication policy controls, and **Add Rule > Add Filter,** click **Authentication Profiles** and display the list of existing profiles and select a profile to use or click **Add New Profile**.

   You can use the same profile for server access, and to re-authenticate for roles and rights that require multi-factor authentication. However, if you want to specify different authentication challenges from which a user can select when executing UNIX commands or accessing Windows applications, select **Add New Profile**.

   As with the Login Authentication Profile, you can select multiple types of authentication to present for the first and second challenges. However, only the authentication challenges that are applicable for a user can be presented when the user attempts to access privileged Windows rights or execute UNIX commands with elevated privileges (dzdo) or in a restricted shell (dzsh).

2. Click **Save**.

You can prepare for multi-factor authentication before or after installing Server Suite components. The steps in this section summarize what to do to finish configuring multi-factor authentication for login access and executing commands for computers in hierarchical zones. You can use Access Manager, adedit, or Access Module for PowerShell scripts to complete most of the next steps.

For more information about performing these tasks, see the following documentation:

- Planning and Deployment Guide
- Administrator's Guide for Linux and UNIX
- Administrator's Guide for Windows

For example, see the *Administrator's Guide for Linux and UNIX* for more detailed information about how to create zones, configure role definitions, and add command rights for Linux and UNIX computers.

To Configure Multi-Factor Authentication

1. Install Access Manager and other components.
2. Create at least one hierarchical zone.
3. Verify the Identity platform instance URL for the zone by displaying the zone properties, then clicking the Platform tab.

   If necessary, you can click Browse to select a different Identity platform instance if you have access to more than one customer-specific Identity platform instance URL.
4. Assign the predefined require MFA for login role definition to the Active Directory users who have access to computers where you want to require multi-factor authentication and who are already assigned the UNIX Login or Windows Login role.

   Alternatively, you can create one or more custom UNIX or Linux role definitions that include the **Require multi-factor authentication** system right. Note that you can also use the Access Module for PowerShell to set the system right described in this step.
5. Define the rights you would like to add to the role and select the **Require multi-factor authentication** reauthentication option on the Attributes tab.

   After you create rights that require multi-factor authentication, add the rights to the appropriate role definitions and assign the roles to the appropriate Active Directory users.

   Note that you can also use the Access Module for PowerShell to require multi-factor authentication for command execution.
6. Refresh the agent.

   For a UNIX computer requiring multi-factor authentication, run `adflush -f` or restart the agent to test multi-factor authentication for login access and command execution.

   For a Windows computer requiring multi-factor authentication, run `dzrefresh` from a command prompt. Depending on your permission settings, you may need to open the command prompt using "Run as administrator."

   Note: When you initially update or install the Server Suite Agent for Windows and configure multi-factor authentication for login, there may be a slight delay while the cache refreshes. During this short period, users who are required to use multi-factor authentication to log in may only be asked for their Active Directory credentials. When they logout from their machine, the cache will have refreshed, and they will then be required to use multi-factor authentication in future login attempts.

After you have completed the basic steps to enable multi-factor authentication, you might want to customize the configuration to suit your environment or to address specific scenarios. For example, you might want to enable group policies or set configuration parameters if you want to modify the default multi-factor authentication operations.

For more information on setting group policies for multi-factor authentication, please see the *Group Policy Guide*. For information on setting configuration parameters, see the *Configuration and Tuning Reference Guide*.

The next sections discuss the most common customization scenarios.

Add Rescue Rights

You should have at least one role with the "rescue" system right for the UNIX and Linux computers in hierarchical zones where you are requiring multi-factor authentication. This system right enables selected users to log in in cases where multi-factor authentication cannot be completed. For example, if a UNIX computer where multi-factor authentication is required is disconnected from the network and cannot access the Delinea Platform, only users with the "rescue" right will be able to log in until the connection to the identity platform is restored.

Configuring Secure Shell (ssh) for Multi-Factor Authentication

If you are planning to require multi-factor authentication for secure shell (ssh) sessions and you want to use a native secure shell package, you should review the settings in the secure shell configuration file (sshd_config) to be sure that the ChallengeResponseAuthentication option is set to yes.

You can edit the file manually or enable the "Allow challenge-response authentication" group policy to automatically configure this setting. You can find this group policy in the Group Policy Management Editor under Computer Configuration > Policies > Centrify Settings > SSH Settings. For more information about adding, enabling, and applying Centrify group policies and the other group policies you can use for secure shell sessions, see the *Group Policy Guide*.

Enforcing Multi-Factor Authentication for Single Sign-on Login Access

If you use the Delinea OpenSSH package, you can require multi-factor authentication for secure shell connections even for single sign-on access to remote computers. In this scenario, users must respond to the authentication challenges to open the secure shell connection then be silently authenticated to additional services and computers. Note that this scenario is only supported if you are using the Delinea version of the OpenSSH package and not supported for native secure shell packages. To enable multi-factor authentication for single sign-on using secure shell sessions, you must enable and apply the Enable SSO MFA group policy. You can find this group policy in the Group Policy Management Editor under Computer Configuration > Policies > Centrify Settings > SSH Settings. For more information about adding, enabling, and applying Centrify group policies and the other group policies you can use for secure shell sessions, see the *Group Policy Guide*.

If you are not enabling and applying group policies for Delinea-managed computers, you can manually enforce multi-factor authentication for single sign-on by setting the secure shell configuration parameter SSOMFA to yes in the /etc/centrifydc/sshd/sshd_config file.

If you enable the group policy or set the parameter and auditing is set to required, users who access a Delinea-managed computer using ssh or PuTTY are prompted to respond to the multi-factor authentication challenges before starting the shell session. Securing the shell session with multi-factor authentication prevents unauthorized users from using the secure shell session to connect to remote services and computers.

Require Multi-Factor Authentication for PAM Applications

If you select the "Multi-factor authentication required" system right in a role definition, the PAM applications you add to the role will require users to provide a secondary form of authentication to log in successfully. You define the forms of authentication available and presented to the user in the **authentication profile** you have configured in the Privileged Access Service using the administrative portal.

Note that some applications do not support multi-factor authentication and users might be denied access to applications that they would otherwise be able to use. For example, if a specific version of an application that you want to use only supports a single layer of authentication—such as a password challenge—users would be prevented from logging on and using the service even if they are assigned to a role with the predefined login-all PAM application right.

If you want to grant access to applications that only support one layer of authentication in roles where you are generally using the "Multi-factor authentication required" system right, you must add those applications to the list of applications for which you want to skip multi-factor authentication. You can update the list of applications for which to skip multi-factor authentication by enabling and modifying the "Specify programs for which multi-factor authentication is ignored" group policy or setting the pam.mfa.program.ignore configuration parameter in the centrifydc.conf file.

Before assigning roles with multi-factor authentication required to users, you should test access to all of the applications you expect users to access to verify

they won't be unexpectedly denied access simply because multi-factor authentication isn't supported. Because the applications that don't support multi-factor authentication will depend on the platforms and the versions of the applications you plan to support, testing in your own environment is the only way to determine which applications to add to the pam.mfa.program.ignore configuration parameter.

The most common applications that are known to only support a single password challenge and response for authentication are ignored for multi-factor authentication by default. For example, some versions of vsftpd, java, and httpd do not support multi-factor authentication and are ignored by default.

Additionally, while some platforms support multi-factor authentication for all PAM applications, they may not allow you to require multi-factor authentication for GUI log in. For example, for users running AIX, Solaris, and HP-UX, multi-factor authentication for GUI login is not supported.

Configure Multi-Factor Authentication in Legacy Zones

If you want to configure multi-factor authentication for UNIX and Linux computers in classic zones or in Auto Zone, you must follow different steps than in hierarchical zones. For multi-factor authentication on computers in the "legacy" types of zones, you must either enable and apply group policies or set configuration parameters.

You can find these group policies in the Group Policy Management Editor under Computer Configuration › Policies › Centrify Settings › DirectControl Settings › MFA Settings. For more information about adding, enabling, and applying Delinea group policies, see the *Group Policy Guide*. For more information about setting configuration parameters, see the *Configuration and Tuning Reference Guide*.

The following sections describe multi-factor authentication configuration options for Server Suite-managed Windows computers. In addition to these options, you can use group policies to customize basic operations for connecting to the Delinea Platform and multi-factor authentication on Windows computers. For more information on these group policies, please see the *Group Policy Guide*.

You can find the group policies for multi-factor authentication and the grace period on Windows computers in the Group Policy Management Editor under **Computer Configuration > Centrify Settings > Windows Settings > MFA Settings**.

To set the grace period, use the following group policies:

- Configure multi-factor authentication lock screen grace period. This group policy enables the grace period for lock screen.
- Configure multi-factor authentication user privilege elevation grace period
  This group policy allows the administrator to set the grace period for privilege elevation for users.

Reset Password

Password reset is a very popular self-service capability for Identity and Access Management solutions: It reduces calls to the help-desk and enables users to become productive quickly. The system allows the user to make a limited number of reset password requests within a specified period.

Note: This feature does not enable the user to unlock their account.

To reset your password (user instructions):

1. On the login screen, click the **Forgot Password** link.

   A prompt appears asking the user name. (If the user already entered their username using the login screen, it appears in this user name field.)

2. Complete the MFA challenges, which are based on the password reset profile.

3. A new prompt asks you to enter a new password and confirm it.

   After resetting the password, you can log in using the normal login screen.

Disable Self-Service Password Reset

Configuring this policy setting allows the administrator to force disable the password reset.

You can use this group policy to allow the administrator to force disabling of the password reset feature. There are two settings for this group policy:

- Enabled: If this policy is set to **Enabled**, the self-service password reset feature on the machine is disabled, including the cloud-enabled self-service password reset.

- If this policy is set to **Disabled** or **Not Configured**, the self-service password reset feature on the machine follows the cloud policy setting (cloud policy settings can be found at: **Policy Settings** › **User Security Policies** › **Self Service** › **Password Reset**). The cloud policy settings are accessed through the Admin Portal.

  Note: The Admin Portal is available after you log in to a Delinea Platform instance.

Configure Offline Multi-factor Authentication and Rescue Users

When a Windows computer that is running a Server Suite Agent is not configured to use multi-factor authentication, you can use a local account to rescue that system when it cannot connect to the Delinea Platform.

Local users do not require MFA challenge. For non-safe mode in Windows Operating System:

- With **Privilege Elevation Service** (zone mode) a local user can log in without MFA whenever they cannot connect to the Centrify Platform. The exception is that the local user does not assign roles for either "Window login permit" or "rescue user."
- Without the **Privilege Elevation Service** (zoneless mode) a local user is allowed to log in without MFA.

For safe mode in Windows Operating System, a local user can only log in as a rescue user. To set a rescue user for zone mode, go through the Access Manager to assign roles. To set a rescue user for zoneless mode, go through Group Policies to configure.

If a computer that is joined to a zone starts in Safe Mode, only users who are assigned a Login role with the system rescue right selected will be able to access the machine. These users will not be required to use multi-factor authentication.

Users who are required to use multi-factor authentication to log in to their Windows workstations can set up an offline MFA profile to use as a second form of authentication in the event that their machine cannot connect to the Delinea Platform. These users will see a system notification urging them to set up this passcode each time they log in to their machine until they configure it.

Users set up their offline MFA profile in following way:

To set up an offline MFA profile:

1. Right click the Delinea notification icon in the system notification area, and select **Setup Offline MFA Profile**.
2. Click **Next** to begin the Offline Authentication Wizard.
3. Select one of the following methods to create a authenticator account profile on your mobile device:
    - **Scan barcode**

      If you select this option, a QR code is displayed for you to scan using your mobile authenticator application. You can use either the Delinea application or a third-party authenticator application.
    - **Manual entry**

      If you select this option, you must manually enter the displayed account profile information into your authenticator application.
    - **Program YubiKey**

      If you select this option, you can use a YubiKey as the second form of authentication. You'll then need to select which slot on the YubiKey to use, and whether or not to use Yubikey's touch-to-sign feature.
4. Enter the passcode that is generated after you have created your authenticator profile. Click **Next**.
5. Click **Finish** to exit the Wizard.

After a user has set up their offline MFA profile, they will be prompted to enter the mobile passcode generated by their authentication application or YubiKey as the second form of authentication when they attempt to log in to their machine if it cannot connect to the Delinea Platform instance.

Note: If you have already set up your offline MFA profile and want to reconfigure (override) it, you will be prompted for multi-factor authentication. That profile is set in the MFA Login Policy.

Require Multi-Factor Authentication using Computer Roles

Computer roles can enable you to group and provide access to computers through role assignments. One strategy you might find useful is to use computer roles to control where multi-factor authentication should apply. For example, you might have several computers with highly sensitive material where you want to ensure all user access will require multi-factor authentication. To accomplish this goal, you can configure a computer role, then add and remove computers with sensitive information to control whether multi-factor authentication is required.

To require multi-factor authentication based on a computer role

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new computer role.
3. Expand Authorization to select Computer Roles, right-click, then click **Create Computer Role**.
4. Type the role name and, optionally, a role description, then select **<Create group>** for the Computer group to create a new Active Directory group for computers.

   For example, to create a new Active Directory security group for the computers with sensitive information, click **Browse** to select the Active Directory location for the new group. If you are using the default deployment structure, you would browse to a location similar to acme.sales.org/ACME/Computer Roles then type a group name such as mfa_required_consols, select a scope, and click **OK**.
5. Click **OK** to save the new computer role.

6. Add the computers that require multi-factor authentication for access to the mfa_required_consoles Active Directory security group.

   As you add computers to the Active Directory security group, the computers are listed as Members of the computer role.

7. Expand the computer role you created in Step 4, select Role Assignments, right-click, then select **Assign Role**.

   For example, if you created a new computer role with the role name CR_MFA_required, expand that computer role name to select Role Assignments, right-click, then select **Assign Role**.

8. Select the predefined require MFA for login role definition, then click **OK**.

9. Select **All Active Directory accounts**, then click **OK**.

Using Multi-Factor Authentication when there are Selective Cross-Forest Trusts

If you have domains in different forests that have a two-way selective trust relationship, any computer or user accounts that are used to log on to the remote forest must be granted the "Allowed to authenticate" right on the domain controllers in both forests to get role information.

In addition to granting the "Allowed to authenticate" right to users and to computers with the Server Suite Agent for Windows installed, the right must also be granted to computers that host your Cloud Connectors. <

After you grant these computers and users the "Allowed to authenticate" right for the domains in both forests, users that are assigned a role with a multi-factor authentication right for login and privilege elevation will be able to authenticate using any of the authentication mechanisms that you have assigned to them.

If a connector is not allowed to authenticate on the remote domain controller, some multi-factor authentication mechanisms may fail to authenticate users.

Configuring MFA with RADIUS for Privilege Elevation Service for Windows Checklist

This document provides a configuration checklist for 3rd party multi-factor authentication providers such as Duo, Okta, SecurID (or any other vendor that provides a RADIUS service) to provide identity validation with the Privilege Elevation Service in the Microsoft Windows platform.

If you have an identity service provider (such as Duo, Okta, SecureID, and so forth) that you use for MFA logins, you can integrate authentication and privilege elevation with your identity provider and the RADIUS protocol to require MFA for privilege elevation tasks, such as Run with Privilege and New Desktop.

Make sure that you work with your RADIUS expert along with your network and directory services lead administrators during the design and configuration tasks.

The checklist below includes links to documented procedures.

Note: If you use Privileged Access Service, although you can enable MFA with RADIUS, the recommended practice is that you use the native integration.

	RADIUS requirements	
1	Gather the following settings for your RADIUS service: IP address or fully qualified domain name Port Timeout settings Pre-shared secret	
2	Verify that you can generate a RADIUS one-time password successfully.	
	Verify that identity	

3	authentication is working correctly with your RADIUS system.	
4	Have access to someone who is knowledgeable about your RADIUS system and can answer questions or help troubleshoot issues, if needed.	
	Windows and Active Directory requirements for RADIUS configuration	
5	A Windows computer to use as a RADIUS client for initial testing, including: Client name Client IP address	
6	Make sure that client systems can reach the RADIUS server over the network (check your firewall settings). You may need help also from your network team if your RADIUS cluster has a load-balancer in the front end.	
7	You have administrative access to the designated Windows computer so that you can install software and do configurations.	
8	You have Active Directory account access so that you can modify group policies that apply to the target computer.	
9	You have access to the Group Policy Management Console.	
10	Your Active Directory expert must decide how the group policy layout and scope will be designed so that the group policies	

	are applied to the clients based on their RADIUS service availability.	
	Authentication and Privilege Elevation Services Requirements for RADIUS configuration	
11	Access Manager console is installed on the client computer.	For details, see "Run the setup program on a Windows computer" in the *Administrator's Guide for Windows*.
12	The Agent for Windows is installed on the client system, you've configured the system to work with Privilege Elevation Service, including joining the computer to a zone.	For details, see "Install agents for Windows" in the *Administrator's Guide for Windows*.
13	You have administrative access to Access Manager so that you can manage roles and rights.	
14	The group policy templates from release 19.6 or later are installed. For RADIUS configuration, you need at least the Centrify Windows settings group policies.	For details, see "Install group policy extensions separately from Access Manager" in the *Administrator's Guide for Windows*.
15	If you want to capture the RADIUS events in your SIEM system, make sure the Audit trail is configured to go to the local log file.	In GPME, go to computer Configuration > Policies > Centrify Audit Trail Settings > Centrify Global Settings > Send audit trail to log file (this is not configured by default). For details, see "Send audit trail to log file" in the *Group Policy Guide*. For details, see "Send audit trail to log file" in the *Group Policy Guide*.
16	You have a role and user to test with. Make sure the role has rights for privilege elevation, such as New Desktop rights or Run as Role.	Make sure that you can elevate privileges successfully for that user and role before you try to configure RADIUS authentication.
	Configure a system to use RADIUS for privilege elevation (using group policies)	
		Configure the following group policies: Windows > MFA Settings > Specify the authentication source for privilege

17	Enable and configure the RADIUS group policies.	elevation : set this policy to RADIUS Authentication. Windows > MFA Settings > Remote Authentication Dial-In User Service (RADIUS) Settings > Enable Remote Authentication Dial-In User Service (RADIUS): enable this policy. Specify the RADIUS connection timeout: Configure to match your RADIUS timeout setting. Specify the RADIUS server IP address: enter your RADIUS IP address. Specify the RADIUS server port number: enter your RADIUS port number (the default is 1812). For details, see "Remote Authentication Dial-In User Service (RADIUS) Service Settings" in the *Group Policy Guide*. For details, see "Remote Authentication Dial-In User Service (RADIUS) Service Settings" in the *Group Policy Guide*. After you update the policies, do a group policy update on the Windows client computer.
18	Configure the role to require re-authentication using multi-factor authentication.	For example: Right-click your test role and choose Properties. The Role Properties dialog box opens. Click the Run As tab. Select Re-authenticate current user and then select Require multi-factor authentication. Click OK to apply the changes.
19	Run dzflush to make sure that the agent has the changes from Access Manager.	For details, see "Using dzflush" in the *Administrator's Guide for Windows*.
20	Set the RADIUS shared secret.	The RADIUS secret is unique to each system and will match the secret that the RADIUS server has. You can set the pre-shared secret by either of the following methods on the client computer: Run the Set-CdmRadiusSecret cmdlet to set the RADIUS shared client secret. For details, see the DirectAuthorize PowerShell cmdlet help. Use the Agent Configuration settings dialog box to configure the RADIUS server, including the pre-shared secret. For details, see "Configuring agent settings for the Identity Services Platform" in the *Administrator's Guide for Windows*.
	TEST AND VERIFY	
21	Verify that a user can elevate privileges by entering the RADIUS one-time password.	For example, if your role has New Desktop rights: Right-click the System Tray and select New Desktop. In the dialog box that appears, select your test role and click OK. If the RADIUS authentication has been configured successfully, you are prompted to enter a password for RADIUS authentication. Enter the password and click Next to continue. You can also view the audit trails for the successful authentication in the system's event log.
22	Verify that a user cannot elevate privileges after entering an incorrect RADIUS one-time password.	

Because multi-factor authentication for Delinea-managed computers relies on the infrastructure of the Privileged Access Service, troubleshooting the configuration of your environment and potential connectivity issues can be challenging. To help you test and verify the proper configuration of an integrated environment, Delinea provides several diagnostic tools.

The following Delinea diagnostic tools are available on Windows computers:

- **Diagnostics Tool.** The diagnostics tool is available through the Agent Configuration service, and is described in Viewing Windows diagnostics.
- **Privilege Elevation Service Diagnostic Information panel** (formerly the DirectAuthorize Agent Control Panel) The information panel is available from the Agent Configuration service, and is described in the *Administrator's Guide for Windows*.
- **The dzdiag command.** The command is available from the Windows command prompt, and is described in the *Administrator's Guide for Windows*.

The following diagnostic tool is available on UNIX and Linux computers:

- **The adcdiag program.** The program is available from the UNIX or Linux command line, and is described in Viewing UNIX and Linux diagnostics.

Viewing Windows Diagnostics

The Server Suite Agent for Windows provides logging and diagnostic services. If you have administrative access on a local computer, you can generate diagnostic information about the operation of the agent for Windows and view and save the current content of the log file from the agent configuration panel. For example, you can generate diagnostic information about user sessions, user roles, desktops, and elevated account access, as well as detailed information about auditing from the agent configuration panel.

There are three different types of diagnostics information available:

- **Centrify Audit & Monitoring Service** provides the diagnostic information related to the auditing and monitoring service.
- **Centrify Identity Platform** provides the diagnostic information related to Privileged Access Service, such as for MFA. This diagnostics tool runs the following tests:
  - **Agent Service Connectivity Check:** Checks to see if the agent is in service, and if the agent is running in a normal state. Also determines whether the agent is in a zone, or is configured to use zoneless mode.
  - **Centrify Connector Connectivity Check:** Determines whether all connectors in the network can be connected properly.
  - **Centrify Identity Platform Certificate Validation Check:** Checks whether the certificates (IWA and cloud) have been installed properly. Also determines whether the agent can be connected without a trusted certificate problem.
  - **Centrify Identity Platform Connectivity Check:** Determines whether a connection to the cloud tenant is functional. Checks for problems with DNS, the firewall, and proxy server settings.
  - **MFA Configuration Check:** Determines whether the local computer has been configured properly. If the computer is in a zone, the test also checks whether MFA complies with the configuration defined in the zone.
  - **MFA Role and Permission Check:** Verifies whether role permissions are set properly in the Privileged Access Service Admin Portal.
  - **Offline MFA Provisioning Check:** Determines if the computer has been configured with an offline MFA profile or not.
- **Centrify Privilege Elevation Service** provides the diagnostic information related to privilege management.

You can view these diagnostics tools either from the Windows system tray or from the agent configuration panel.

For more details, see the Administrator's Guide for Windows.

To view diagnostics from the Windows system tray:

1. Log on to a computer where the Agent for Windows is installed.
2. In the Windows system tray, right-click the Centrify icon and click **Troubleshooting**, then select the service for which you want to view diagnostic information (your options may vary depending on what services are enabled on the computer):
   - **Centrify Audit & Monitoring Service** opens a dialog box with a text-based summary of diagnostic auditing and monitoring information.
   - **Centrify Identity Platform** runs a series of connectivity tests and lists out the results of each test.
   - **Centrify Privilege Elevation Service** opens a dialog box with a text-based summary of diagnostic privilege elevation information.

To generate diagnostics or view the log file from the agent configuration panel:

1. Log on to a computer where the Agent for Windows is installed.
2. In the list of applications on the Windows Start menu, click **Agent Configuration** to open the agent configuration panel.
3. Select the service for which you want to view information:
   - **Centrify Audit & Monitoring Service** opens a dialog box with a text-based summary of diagnostic auditing and monitoring information.

- **Centrify Identity Platform** runs a series of connectivity tests and lists out the results of each test.
- **Centrify Privilege Elevation Service** opens a dialog box with a text-based summary of diagnostic privilege elevation information.

4. Click **Settings**.
5. Click the **Troubleshooting** tab.
6. Click **Diagnostics** to generate diagnostic information.
7. Select the Diagnostic Information displayed, right-click, then select **Copy** to copy and paste the output to a file for further analysis.
8. Click **View Log** to display the current log file for the local agent.
9. Click **Options** to see or change the location of the log file or the level of detail recorded in the log file.

The adcdiag program performs a set of tests to check for access to a Delinea server authentication instance, the availability of one or more connectors, whether the computer is joined to an Active Directory domain, and whether the connector you are attempting to use is configured to use integrated Windows authentication.

To perform the set of tests to verify a UNIX or Linux computer can be configured to use multi-factor authentication, run the following command:

/usr/share/centrifydc/bin/adcdiag

By default, the command displays the test results in standard output (stdout) and generates a diagnostic report in the /var/centrify/tmp directory with a dated time stamp similar to the following:

adcdiagCheckingReport_20160307_151128.log

If any of the tests returned errors or warnings, you can check the diagnostic report for additional information, including suggestions for resolving any issues found. For details about the command-line options available for the adcdiag command, see the man page for the command.

Depending on how your Windows environment is set up, you may have to specify a trusted host certificate in order to enable multi-factor authentication. If you do not do this, you will see an error message during installation and configuration.

In a production environment, it is strongly recommended that you specify an existing trusted host certificate from a known third-party certificate authority, such as GoDaddy or Verisign. Using a self-signed certificate in a production environment can leave your environment vulnerable to security breaches.

For details about importing a trusted host certificate, see To import the certificate manually to a local Windows computer.

Privileged Access Service cannot manage the password for a user if multi-factor authentication is required for the user to log in. You can still add a multi-factor authentication-required user account to a PAS resource – with “Manage this password” unchecked - to log in from PAS. However, you may see the status as “Failed” due to system delay. If the operation is successful, then no status will be shown for this user.

If you have installed the agent and enabled the privilege elevation service and users can't log in for some reason, try to log in to the agent system in either Windows Safe mode or rescue mode.

We also recommend that you assign some users to the "Rescue - always permit login" role; that way, they can still log in even if the agent providing MFA isn't working for some reason. For details, see Configuring offline multi-factor authentication and rescue users.

Typically, a Unix/Linux system running the Server Suite Agent for *NIX is located on a private network. By default, the agent uses the Cloud Connector as a HTTP proxy server for connecting to the Cloud Connector to, for example, perform multi-factor authentication.

However, you may prefer to reconfigure the agent to use a different proxy server. The following sections explain how to do that.

Requirements

The configuration described in these sections require the following conditions:

- The HTTP proxy server is used only for communication between the agent and the Delinea Platform. It does not provide proxy services for another purpose.
- All of the Agents must use the same proxy server.
- You have installed DirectControl on the system.

Also note the following points:

- All of the agents use a single proxy server configuration (multiple configurations not supported)
- The machine password keytab must contain at least two versions of the key.
- This proxy server configuration supports all zone types.
- The information in the following sections does not apply to Mac OS X.

Configure the Agent to Use a Custom HTTP Server

To configure the Agent to use a custom HTTP server, use the following command syntax to set the custom HTTP proxy server:

adwebproxyconf --set,-S [--username,-u <username>]
[--password, -p <password>] [--machine,-m]
[--server,-s <servername:portI>*]*
[--authreq,-r <truelfalseI>] [--authtype, -t <type>],
'--version,-v','--help,-h' and '--verbose,-V'

For example, enter the following command (replacing the angle brackets and placeholders with actual values):

adwebproxyconf --set --username <username>
--password <password> --server <servername:port>
--authreq <true> --authtype <type>

Change the value of adclient.cloud.direct.connection to false.

Verify that the new configuration works. Enter:

adwebproxyconf --test --cip <cip url>
--server <servername:port>

For more information, see Command Reference.

HTTP Proxy Credential Local Storage

This section describes how the HTTP proxy credentials are stored locally on the Unix/Linux system that's running the agent.

The HTTP proxy credentials are be stored only in the local kset file: /var/centrifydc/httpproxy.cred

This httpproxy.cred file is only readable and write-able by root.

For security, remove httpproxy.cred. from the system when you remove the system from the domain.

For security, the proxy user's password is encrypted before being stored in httpproxy.cred.

Password Encryption

The proxy user's password is encrypted using the system's principal key, which is normally stored in /etc/krb5.keytab.

It should use the latest key to do encrypt the password. By default, it uses AES256-CTS-HMAC-SHA1-96 encryption.

If the key for a particular encryption type is not available, the Agent uses the next preferred and available encryption type that has a key in the system's keytab file.

When the system password changes, the agent uses it to re-encrypts the proxy server password. The system keytab file keeps the two latest versions of key.

If the Agent on the Unix/Linux system has FIPS Mode enabled, only a FIPS-compliant encryption type is allowed to encrypt the proxy credential password.

If a password is encrypted with non-FIPS-compliant encryption type, even if the machine keytab contains a valid key, the agent will not be able to decrypt it. If that happens, set the proxy password again so that it is encrypted using a FIPS-compliant encryption type.

Encrypted Password Storage

The encrypted password and relevant information is represented in ASN.1 as shown below and is encoded using ASN.1 Basic Encoding Rule (BER) as defined in Section 5.1 of the RFC 4511 LDAP Protocol (https://www.ietf.org/rfc/rfc4511.txt):

```
username STRING,
kvno UInt32,
etype Int32,
cipher OCTET_STRING
}
Int32 ::= INTEGER (-2147483648..2147483647) -- signed values re-presentable in 32 bits
UInt32 ::= INTEGER (0..4294967295) -- unsigned 32 bit values
```

Where:

- username: The proxy user's name.
- kvno: The version number of the key under which the data is encrypted
- etype: The encryption type used to encrypt the cipher. The encryption type number MUST be a type that is supported by the Kerberos protocol.
- cipher: The encrypted password

Local Machine Account Support

In some cases, the current system account's Kerberos credentials should be configured, the username be S-1-5-18, and the cipher part must contain an octet string with 0 length.

Command Reference

adwebproxyconf

The adwebproxyconf command configures the HTTP proxy server settings and credentials on the local system. Typical use cases are:

- Set up the HTTP proxy credential to be used by agent.
- Delete the HTTP proxy credentials.
- Get information about the HTTP proxy credentials.
- Test the proxy connection using the configured credentials
- Test the proxy connection using the supplied credentials

Requirements

- Only root can run this command.
- To run, the system must have joined a zone.

Synopsis

```
adwebproxyconf --set,-S [--username,-u \<username\>]
 [--password, -p \<password\>] [--machine,-m]
 [--server,-s \<servername:port\|""\>]
 [--authreq,-r \<true\|false\|""\>]
 [--authtype,-t \<basic\|digest\|ntlm\|negotiate\|anyauth\|""\>],
 '--version,-v','--help,-h' and '--verbose,-V'
 adwebproxyconf --delete,-D
 adwebproxyconf --list,-L
```

adwebproxyconf --test,-T \<--cip,-c \<cip url\> \>
[--server, -s \<servername:port\>]
[--username,-u \<username\>]
[--password, -p \<password\>] [--machine,-m]

Command Options

--set,-S

Set the HTTP proxy server and credentials for the local system. After using adwebproxyconf -S, use adreload to force the agent process (adclient) to reload its configuration files.

the configuration properties in the /etc/

centrifydc.conf file and in other files in the /etc/

centrifydc directory.

--delete,-D

Delete the HTTP proxy credentials from the local computer and reset the HTTP Proxy configurations in centrifydc.conf:

- HTTP Proxy Server
- HTTP Proxy authentication type
- HTTP Proxy authentication required

--list,-L

List the HTTP proxy username and server from the configuration on the local system.

--test,-T

Test the HTTP proxy credential using configured or supplied settings.

--username,-u ‹username›

Proxy username. If a username is not supplied but --password is supplied, the username defaults to Administrator'

--password, -p ‹password›

Proxy user's password, if not provided, will be prompted

--machine,-m

Use local machine account for proxy authentication, and SPNEGO authentication is used.

This option cannot be used with -u, -p, or -t.

--authreq,-r ‹truelfalsel""›

Specify if HTTP Proxy authentication is required in centrifydc.conf, Optional.

Can use individually or with other options.

--authtype,-t ‹basicldigestlntlmlnegotiatelanyauthl""›

Specify if HTTP Proxy authentication authentication type in centrifydc.conf. Optional.

Please refer to above section for valid values.

Can use individually or with other options.

--server, -s ‹servername:portl""›

Specify HTTP Proxy Server to use to update to centrifydc.conf. Optional.

Empty string unsets the value in centrifydc.conf.

Can use individually or with other options.

--version,-v

Specify the version.

--help,-h

Get command line help.

--verbose,-V

Get additional details while the settings are being applied.

--cip,-c <cip url>

Specify the URL of the identity platform.

Must be specified.

--server, -s <servername:port>

Specify HTTP Proxy Server to test.

If not specified, get it from centrifydc.conf

--username,-u <username>

Proxy username.

If not specified, get it from proxy cred file.

--password,-p <password>

Proxy user's password.

If username is specified, but password is not, prompt for it.

If both username and password are not specified, get it from proxy cred file.

--machine,-m

Use local machine account for proxy authentication, and SPNEGO auth type is used.

This option cannot be used with -u, -p, or -t.

This test option always use the proxy credential to check the connection to CIP thru the specified HTTP Proxy server.

You can also use the adcdiag command to check HTTP proxy server settings.

If you have an identity service provider (such as Duo, Okta, SecureID, and so forth) that you use for MFA logins and you've integrated that with Server Suite and you have a RADIUS server for MFA with your identity provider and also a RADIUS server integrated with Server Suite, you can set up silent authentication so that your users don't have to enter their passwords or security question answers twice.

You can configure the credential provider to silently send the user's password as the first response to the authentication workflow. This feature prevents prompting the user for password multiple times when a 3rd party radius is being used as the authentication mechanism.

You can deploy these registry settings as group policies.

To control this new feature we have 3 new registry settings:

- **SilentAuthPromptDetectionRegex** (string)

  This is a regular expression that we match against an authentication prompt that RADIUS sends us. If there is a match, we'll try to automatically respond. For example, if the RADIUS prompt is "Enter your password" we can set this regular expression to

  .*password.*

- **SilentAuthPromptResponseType** (uint):

  This setting controls the kind of response we provide.

  0 - (default) No auto-response: The service prompts the user for the password even if there's a regular expression match.

  1 - Silent auto-response: The service responds with the same password that the user just entered as a Windows login credential.

  2 - Fixed response: Instead of responding automatically with the password, we respond automatically with a fixed response (a static string, for example "This is an automatic response"). Use the **SilentAuthPromptFixedResponse** to store the fixed response text.

- **SilentAuthPromptFixedResponse** (string):

  The fixed response if the **SilentAuthPromptResponseType** is configured as 2. Use this setting to store the fixed response, such as "This is an automatic response."

When you create these registry entries, put them in HKEY_LOCAL_MACHINE\SOFTWARE\Centrify\DirectAuthorize\Agent.

If you are an administrator of Server Suite-managed UNIX or Linux computers, you can use this guide to help you set up a Certificate Authority with the Microsoft Windows certificate auto-enrollment feature to automatically manage certificates for UNIX and Linux computers in your domain. While there are many ways to deploy certificates, this guide describes how to use Active Directory server roles and Windows Group Policy to set up automatic enrollment.

This area includes the following topics:

The Server Suite Agent uses the Microsoft Windows public key infrastructure (PKI) to obtain the certificates used by your Server Suite-managed UNIX or Linux computers that are joined to a domain. By joining to the domain, these computers become eligible for auto-enrollment.

The most basic configuration of the Windows PKI environment utilizes a Windows server as the Certificate Authority (CA) that issues and manages security credentials and public keys through the exchange of encrypted digital certificates. The Server Suite Agent then uses the Microsoft Windows certificate auto-enrollment feature of the Certificate Authority to make certificates available to UNIX computers.

This section describes how to set up a basic environment that has a single, enterprise root Certificate Authority (CA). In this scenario, the Certificate Authority is a Microsoft Enterprise Certificate Server that issues all certificates. In a production environment, you may have more complex requirements that include multiple CAs configured for a domain. However, setting up this sample environment should give you enough information to extend your PKI configuration to a more complex environment.

The Server Suite Agent requires a Microsoft Windows Server to be configured as the Certification Authority (CA) for the Active Directory forest. Additionally, auto-enrollment is not supported for certificates issued by other public or private Certificate Authority services or organizations.

The first step in configuring the environment is to identify a computer to be the Certificate Authority server for the Active Directory forest. This computer must be connected to a network with a server that has Windows Server 2008 (or later) Domain Name Service installed, and it must be joined to the Active Directory domain. In most cases, the computer designated to be the CA should not be a domain controller in a live production environment. To configure the computer as a Certificate Authority, you must install Microsoft Internet Information Services (IIS) and Certificate Services.

Microsoft Internet Information Services (IIS) are required to handle Certificate Revocation List (CRL) requests made by the authentication service and to provide the virtual directories required to issue and manage certificates.

Certificate Services are required to enable the computer to act as a Certificate Authority (CA) and issue certificates to other computers that join the domain. The Application server role, which installs IIS, and the Certificate Services server role must be on the same computer. Therefore it is recommended that you install IIS at the same time you install Certificate Services.

What's Required to Install Certificate Services

Before installing Certificate Services, check that you have the following:

- Account credentials for an account that is an Enterprise Administrator and a Domain Administrator of the forest root domain of the Active Directory forest.
- A computer with Windows Server 2008 Enterprise Edition or later. Previous versions of Windows Server do not support auto-enrollment within the certificate templates. In addition, the computer must be running Enterprise Edition because Standard Edition does not support the V2 or V3 certificate templates that are required for auto-enrollment.
- Active Directory services must be installed on the Certificate Services server. If you install the Certificate Services server role on a domain controller, no further action is required. When you promote a computer to be a domain controller, the Active Directory services are installed automatically.

  Note: This guide details how to configure auto-enrollment on a computer running Windows Server 2012 R2. For information on configuring auto-enrollment for computers running other versions of Windows Server, please visit the Microsoft website.

Adding the Required Server Roles to Make the Computer a Certificate Authority

After you have verified that you have an appropriate account and computer configuration, you can use Server Manager to add the appropriate server roles.

To install IIS and Certificate Services on a Windows Server

1. Open the Server Manager Dashboard and click **Add Roles and Features**.

   Click **Next**.

2. For Installation Type, select **Role-based or feature-based installation**, then click **Next**.

3. Ensure that **Select a server from the server pool** is selected and highlight the server on which you would like to install roles and features. Click **Next**.

4. Select **Active Directory Certificate Services**, then click **Add Required Features** in the pop-up window.

   Click **Next**.

5. Click **Next** to accept the default selections for Select Features.

6. Click **Next** on the notification that you will be unable to change the domain settings after installing Certificate Services.

7. Select **Certification Authority** and click **Next**.

8. Click **Install**.

After Windows restarts, you will see a new Role in Server Manager called AD CS. In the following procedure, you will configure this role to allow your server to act as a Certification Authority.

Configuring the Certificate Authority

1. Click the notification icon in the Server Manager command bar to open the **Add Roles and Features Wizard**.
2. Click the link, **Configure Active Directory Certificate Services on the destination server**.
3. In the AD CS configuration screen, verify that you are logged on as an administrator and click **Next**.
4. Select **Certification Authority** and click **Next**.
5. Select **Enterprise CA** and click **Next**.

   Note: You must be a member of both the Enterprise Admins group and the Domain Admins group to configure an Enterprise Certificate Authority.
6. Select **Root CA** and click **Next**.
7. Select **Create a new private key** and click **Next**.
8. Accept the defaults for the cryptographic provider, key length, and hash algorithm. Click **Next**.
9. Enter a name for the Certificate Authority or accept the defaults, and click **Next.**.

   Note: After the Certificate Authority is configured, you will not be able to change the name.
10. Specify the validity period of the certificate, click **Next**.
11. Accept the default location for the certificate database and click **Next**.
12. Review your CA configuration and click **Configure**.
13. Click **Close** when the confirmation message appears, and restart the server to retrieve a certificate from the CA.

You can use the certificate snap-in to make a copy of a certificate to use on another computer, or to create a backup copy.

In order to establish a chain of trust for your PKI environment, you identify the copy of the CA you just created as a trust anchor.

To establish the CA as a trust anchor, add the root certificate for the CA to the **Trusted Root Certification Authorities** container in the group policy object that defines the IP Security policies.

To Add a Trusted Root Certificate to the Group Policy Object

1. Open the Certificates (MMC) snap-in.

   If the Certificates snap-in is not available, you can run MMC and click **File > Add/Remove Snap-in** to add it.

2. Select Computer account, and click **Next**.
3. Select Local computer, then click **Next**.
4. Click **Certificates > Trusted Root Certification Authorities > Certificates**.
5. Select the root certificate generated by the CA you created in the previous procedure, then double-click it to see its Properties page.
6. Click the **Details** tab; then click **Copy to file** to start the Certificate Export Wizard. In the wizard, make the following selections:
   - **File format**: *DER encoded binary X.509 (.CER)*
   - **File Name**: Anywhere on the local server
   - **Include all certificates in the certification path**: *No*
7. Open the Group Policy Object Editor and select the group policy object that defines the IP Security policies.

   Click **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Trusted Root Certification Authorities**.

8. Select **Trusted Root Certification Authorities**, right click, and select **Import** to open the Certificate Import Wizard.
9. Click **Next** on the **Welcome** screen.
10. Browse to find the root certificate you copied in Step 6, then click to accept the default values on each screen.
11. Click **Finish** to complete the wizard.

The root certificate is now in the Active Directory Trusted Root Certification Authorities container. Certificates in this container are downloaded to any computer that joins the domain to establish trust for the root CA.

The Server Suite Agent uses the Microsoft Windows certificate auto-enrollment feature to make certificates available to UNIX computers. If auto-enrollment is enabled, when a UNIX computer joins a domain, the Server Suite Agent requests certificates from the CA based on particular templates, and installs them on the joined computer.

To enable auto-enrollment, you must do the following:

- Enable auto-enrollment for the group policy.
- Create a certificate template with auto-enrollment enabled.

Enabling Auto-Enrollment for the Group Policy

To enable auto-enrollment for the group policy:

1. Open the Group Policy Management Editor and select the group policy object that defines IPsec policies.

   Click **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Certificate Services Client - Auto Enrollment**.

2. Double-click **Certificate Services Client - Auto-Enrollment**, select Enabled, and check the following boxes:

   - **Renew expired certificate, update pending certificates, and remove revoked certificates**
   - **Update certificates that use certificate templates**

3. Click **OK** to save the auto-enrollment settings.

Creating a Certificate Template

To configure a template with auto-enrollment:

1. Open the MMC Certificate Template snap-in.

   Another way to open the Certificate Template console is to open the Certification Authority console, right-click **Certificate Templates**, and select Manage.

2. Select a template, then right-click and select **Duplicate Template** to create a new template that you can modify.

   For example, select the Workstation Authentication template.

3. On the Properties page for the new template, do the following:

   1. Select the **General** tab and enter a name for the template.
   2. Select the **Security** tab and select **Domain Computers**. Then select **Read** and **Autoenroll** permissions.
   3. Select the **Subject Name** tab. For **Subject name format**, select **Fully distinguished name**.
   4. Select the **Extensions** tab. Then select **Application Policies**.
   5. Click **Edit**. **Client Authentication** should already be shown.
   6. Click **Add**, then scroll and select **Server Authentication**.
   7. Click **OK**.

4. Click **OK** to save the new template.

You can now assign the newly created template to the Certificate Authority. Whenever a computer joins the domain, the CA issues a certificate based on the template, and the Server Suite Agent downloads the certificate to the computer.

To assign the template to a CA:

1. Open the Certification Authority console.
2. Click **Certification Authority** › *CA_name* › **Certificate Templates**, where *CA_name* is the container for the CA you set up earlier in Preparing a computer to be a Certificate Authority (CA).
3. Right-click and select **New** › **Certificate Template to Issue**. Select the template you just created and click **OK**.

The root CA is now set up to issue certificates based on the template you created.

Delinea

Generating a certificate revocation list (CRL) is the method a Certificate Authority (CA) uses to maintain the validity of the certificates that it issues. A CRL contains a list of certificates (or more specifically, a list of serial numbers for certificates) that have been revoked or are no longer valid, and therefore should not be relied upon. The agent retrieves CRLs from CAs after specific events (such as joining a domain) and at specific intervals to determine which certificates, if any, have been revoked, and thus whether to request new certificates.

> **Note**: The current version of the Server Suite Agent only supports complete certificate lists, not delta CRLs, which only describe the updates since the complete list was published.

Generating a Certificate Revocation List (CRL)

A CRL is generated by a CA and contains a list of certificates to revoke from the list of certificates that the CA has issued.

Typically, a CA automatically generates a CRL at a specified interval, anywhere from two hours to one year, at which point the new CRL with the list of revoked certificates is available for clients to request.

The CRL itself contains the interval period, which allows clients, such as Server Suite Authentication Service, to determine when to request a new CRL. See Retrieving a certificate revocation list and verifying certificates for information about retrieving CRLs.

In addition to automatic generation of a CRL, an administrator can use specific Active Directory utilities that allow them to manually revoke certificates and publish a CRL on the CA. In this case, the CRL-publishing interval is reset so the next automatic publishing operation will occur in the appropriate amount of time.

Retrieving a Certificate Revocation List and Verifying Certificates

At specific times (when the UNIX system joins a domain, the administrator issues the adgpupdate command, or the group policy refresh interval occurs), the Server Suite Agent performs certain tasks, including determining whether to retrieve a CRL (Certificate Revocation List). Specifically, the agent does the following:

- Identifies the CA that issued certificates for the system.
- Looks at the refresh interval in the current CRL to determine whether to retrieve a new CRL.
- If the interval has expired, retrieves a new CRL by using the IIS Web Server for the CA.
- Verifies the currently issued certificates against the CRL and requests new certificates for certificates that have been revoked.

  > **Note**: When you manually revoke a certificate, it is possible that the certificate will appear as valid even after running the adgpupdate command to trigger an IPsec update. When you revoke a certificate, the Server Suite Agent first looks at the current CRL to determine the validity of the certificates that have been issued. In this case, the newly revoked certificate still appears as valid. Immediately afterwards, because of the IPsec update, the agent requests a new CRL. The new CRL shows that the certificate in question is invalid, but the agent will not look at the new CRL until the next scheduled update, or until you run the adgpupdate command again. Therefore, to be certain to have current information, if you manually revoke certificates, you can issue the adgpupdate command twice in sequence.

Delinea

Reports and Events

- Reporting Admin Guide
- Audit Events Admin Guide

This guide is for individuals who need to extract audit event information from UNIX and Linux syslogs and Windows application event logs. Additionally, this information is available in the Audit Analyzer. Audit events are organized into categories in the Audit Analyzer and these categories are identified in this document.

Depending on your environment and role as an administrator or auditor, you may want to read portions of this guide selectively. This guide provides the following information:

- Overview of Centrify Server Suite Audit Events provides an overview of how to read audit events.
- Centrify Server Suite Audit Events identifies the different audit event categories. Each audit event includes a sample log with an explanation of how to read the log as well as a list of the available audit events.

Overview of Delinea Server Suite Audit Events

To familiarize yourself with the elements of audit event logs, read the explanations of Windows and UNIX/Linux audit events, and then review how to read Server Suite audit event data.

- Windows and UNIX/Linux Audit Events
- How to Read Audit Event Data
- Configuring the Audit Event Log Location
- Which Events Only in Centrify Audit & Monitoring Service

Windows and UNIX/Linux Audit Events

Review the following examples to understand the Windows and UNIX/Linux audit event logs, and then review how to read audit event data to understand the similarities and differences.

Windows Audit Event Log Line Example

The following is an example of a Centrify audit event recorded in the Windows application event log. Standard Windows audit event fields (in black) contain information about the Centrify event. Centrify augments these standard fields with additional data (in red) to help you to track logon and privilege activity data.

```
04/05/2016 02:15:37 PM LogName=Application
SourceName=Centrify AuditTrail V2 EventCode=6003
EventType=4 Type=Information
ComputerName=member.acme.vms User=NOT_TRANSLATED
Sid=S-1-5-21-3789923312-3040275127-1160560412-500
SidType=0 TaskCategory=%1 OpCode=Info RecordNumber=51645
Keywords=Classic Message=Product: Centrify Suite Category:
DirectAuthorize - Windows Event name: Remote login success
Message: User successfully logged on remotely using role
'ROLE_Windows_Local_Accounts/Global'.
Apr 05 14:15:37 member.acme.vms dzagent[1496]: INFO AUDIT_TRAILlCentrify
SuitelDirectAuthorize - Windowsl1.0l3lRemote login successl5luser=
administrator@member.acme.vms userSid=S-1-5-21-
3789923312-3040275127-1160560412-500 sessionId=6 CentrifyEventID=6003
DAInst=AuditingInstallation DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
role=ROLE_Windows_Local_Accounts/Global
desktopguid=a16f50d8-179b-4d47-93ed-14c10ca76d63
```

Windows Audit Event Log Line Information

The following table provides definitions for each field type and name with their associated field value for the previous example.

Windows Audit Event Log Line Information

Syslog header fields	Timestamp	Apr 05, 2016 02:15:37 PM
	Host Name	member.acme.vms
	Process Name	dzagent
	Process ID	1496
	Log Level	INFO
Centrify audit event header fields	Event Type	AUDIT_TRAIL
	Product	Centrify Suite
	Category	privilege elevation service - Windows
	Product Version	1.0
	Event ID	3
	Event Name	Remote login success
	Severity	5
Centrify audit event common fields for Windows	user	administrator@member.acme.vms
	userSid	S-1-5-21-3789923312-3040275127-1160560412-500

	DAInst	AuditingInstallation
	DASessID	c72252aa-e616-44ff-a5f6-d3f53f09bb67
	sessionId	6
	CentrifyEventID	6003
Centrify audit event-specific fields	role	ROLE_Windows_Local_Accounts/Global
	desktopguid	a16f50d8-179b-4d47-93ed-14c10ca76d63

UNIX/Linux Audit Event Log Line Example

The following is an example of a UNIX/Linux audit event. Centrify audit event information is highlighted in red.

```
Apr 4 21:04:15 engcen6 adclient[1749]: INFO
AUDIT_TRAIL|Centrify Suite|Centrify sshd|1.0|100|SSHD granted|5|user=
dwirth(type:ad,dwirth@acme.vms) pid=7456 utc=1459784055479
CentrifyEventID=27100DAInst= AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6 -d3f53f09bb67 status=GRANTED
service=ssh-connection tty=/dev/pts/0 authMechanism=keyboard-interactive client=
192.168.81.11 sshRights=shell command=(none)
```

UNIX/Linux Audit Event Log Information

The following table provides definitions for each field type and name with their associated field value for the previous example.

UNIX/Linux Audit Event Log Information

Syslog header fields	Timestamp	Apr 4 21:04:15
	Host Name	engcen6
	Process Name	adclient
	Process ID	1749
	Log Level	INFO
Centrify audit event header fields	Event Type	AUDIT_TRAIL
	Product	Centrify Suite
	Category	Centrify sshd
	Product Version	1.0
	Event ID	100
	Event Name	SSHD granted
	Severity	5
Centrify audit event common fields	user	dwirth(type:ad,dwirth@acme.vms)

	pid	7456
	utc	1459784055479
	CentrifyEventID	27100
	DAInst	AuditingInstallation
	DASessID	c72252aa-e616-44ff-a5f
	service	ssh-connection
Centrify audit event-specific fields	tty	/dev/pts/0
	authMechanism	keyboard-interactive
	client	192.168.81.11
	sshRights	shell
	command	(none)

How to Read Audit Event Data

The following information can help you understand how to read Centrify audit events.

Event ID/CentrifyEventID

Every Windows and UNIX/Linux audit event includes two numeric IDs that describe the event. The Event ID in the header fields identifies the unique ID of the event within a particular event category, whereas the CentrifyEventID in the common fields identifies the unique ID among all Centrify audit event types.

Windows Example

	Product Version	1.0	
	Event ID	3	
	Event Name	Remote login success	5
Centrify audit event common fields	user	administrator@member.acme.vms	
	userSid	S-1-5-21-3789923312-3040275127-1160560412-500	
	DAInst	AuditingInstallation	
	DASessID	c72252aa-e616-44ff-a5f6-d3f53f09bb67	
	sessionId	6	
	Centrify EventID	6003	

UNIX/Linux Example

	Product	Centrify Suite
	Category	Centrify sshd
	Product Version	1.0
	Event ID	100
	Event Name	SSHD granted
	Severity	5
Centrify audit event common fields	user	dwirth(type:ad,dwirth@acme.vms)
	pid	7456
	utc	1459784055479
	Centrify EventID	27100
	DAInst	AuditingInstallation

	DASessID	
c72252aa-e616-44ff-a5f6-d3f53f09bb67		
	status	GRANTED
	service	ssh-connection

Severity

Severity is defined by an integer from 0 - 10, with 10 being the most important level. Centrify events are typically a Severity 5.

Spacing

A field name is one word (no spaces) in the audit event file. When the file is processed into a readable format, spaces are added to field names. For example, if you need to search for Management Database Property, you should search on the following term: managementdatabaseproperty.

Case-Insensitive Field Names

Use case-insensitive field names in all search filters.

Configuring the Audit Event Log Location

You can configure audit event logs to go to DirectAudit or your system's default logging system (Windows event log or UNIX syslog). You configure the log location either manually for each computer or by way of group policy.

You can also configure a global audit event logging behavior or specify different settings for different feature areas.

Configuring the Audit Event Logging Location by Group Policy

Audit trail group policies are located in category-specific subfolders (such as **Audit Analyzer Settings**, **Audit Manager Settings**, and so on.

Additionally, a **Centrify Global Settings** subfolder contains group policies that you can set at a global level.

Any category-specific audit trail targets that you set (for example, **Audit Manager Settings** › **Send audit trail to log file**) override global audit trail targets (for example, **Centrify Global Settings** › **Send audit trail to log file**). Each subfolder in **Centrify Audit Trail Settings** contains the same set of group policies.

> **Note**: To send audit trail events to both the database and the local logging facility, enable both of these group policies.

Send Audit Trail to Audit Database

Enable this group policy to specify that audit events for this component **Audit Analyzer**, **Audit Manager**, and so on are sent to the active audit store database.

See the **Explain** tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Send Audit Trail to Log File

Enable this group policy to specify that audit events for this component such as **Audit Analyzer**, **Audit Manager**, and so on are sent to the local logging facility (syslog on UNIX systems, Windows event log on Windows systems).

See the **Explain** tab in the group policy for details about which parameter each group policy sets in the agent configuration file.

Set Global Audit Trail Targets

Specify the target for audit trail information.

If you set this group policy to **Not configured** or **Disabled**, the destination of audit trail information depends on which version of DirectAudit is installed. If DirectAudit 3.2 or later is installed, audit trail information is sent to the local logging facility and DirectAudit. If a DirectAudit version earlier than 3.2 is installed, audit trail information is only sent to the local logging facility.

If you set this group policy to **Enabled**, you can specify the target for audit trail information. Possible settings are:

- 0 (Audit information is not sent.)
- 1 (Audit information is sent to Centrify Audit & Monitoring Service. This capability is supported by DirectAudit version 3.2 and later.)
- 2 (Audit information is sent to the local logging facility, either syslog on UNIX systems or Windows event log on Windows systems.)
- 3 (Audit information is sent to both DirectAudit and the local logging facility.)

This group policy modifies the audittrail.targets setting in the agent configuration file.

Which Events are Only in Centrify Audit & Monitoring Service

Audit events may come from Centrify Authentication Service, Centrify Privilege Elevation Service, or Centrify Audit & Monitoring Service. If you are using only authentication and privilege elevation, the following events will not be available to you as they are from audit and monitoring service:

- All the audit events from the following categories:
  - Audit Analyzer
  - Audit Manager
  - Command
  - Centrify Audit & Monitoring Service - Windows
  - Centrify Audit & Monitoring Service System Management
  - Centrify Audit & Monitoring Service UNIX Agent
  - Centrify Audit & Monitoring Service advanced monitoring
- The following audit events from the category Centrify Commands
  - Auditing enabled (Centrify Event Id 18000)
  - Auditing not enabled (Centrify Event Id 18001)
  - Auditing disabled (Centrify Event Id 18100)
  - Auditing not disabled (Centrify Event Id 18101)

Server Suite Audit Events

This section includes the following topics:

Audit Analyzer

The Audit Analyzer console is a graphical user interface that administrators can use to query and review captured user sessions. The Audit Analyzer is available with the Centrify Audit & Monitoring Service. The Audit Analyzer events focus on session modification.

Audit Analyzer Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 3001. This log sample documents a session being deleted. The change was made by user=administrator@acme.vms on April 20, 2016 at 05:51:01.

```
04/20/2016 05:51:01 PM LogName=Application
SourceName=Centrify AuditTrail V2 EventCode=3001
EventType=4 Type=Information ComputerName=
member.acme.vms User=NOT_TRANSLATED Sid=S-1-
5-21-3883016548-1611565816-1967702834-500 SidType=0
TaskCategory=%1 OpCode=Info RecordNumber=60622
Keywords=Classic Message=Product: Centrify Suite Category:
Audit Analyzer Event name: Delete session Message: 1 out
of 1 selected sessions are successfully deleted. Apr 20
17:51:00 member.acme.vms mmc[4064]: INFO
AUDIT_TRAILICentrify SuitelAudit Analyzerl1.0l1IDelete
sessionl5luser=administrator@acme.vms
userSid=S-1-5-21-3883016548-1611565816-1967702834-500
sessionId=11 CentrifyEventID=3001 DAInst=
AuditingInstallation DASessID=c72252aa-e616-44ff-a5f6-
d3f53f09bb67 sessions_deleted=1 sessions_selected=1
```

Audit Analyzer Audit Events

Audit Analyzer Audit Events

3001	Delete session	Sessions_Deleted: Sessions_deleted Sessions_Selected: Sessions_selected
3002	Delete session by criteria	Delete_criteria: Delete session selection criteria Sessions_Deleted: Sessions_deleted Sessions_Selected: Sessions_selected
3003	Set session reviewers succeeded	Installation: Name of the installation Session Id: Unique identifier of the session Reviewers: List of reviewers of the session
3004	Set session reviewers failed	Installation: Name of the installation Session Id: Unique identifier of the session Reviewers: List of reviewers of the session Reason: Error message
3005	Remove session reviewers succeeded	Installation: Name of the installation Session Id: Unique identifier of the session
3006	Remove session reviewers failed	Installation: Name of the installation Session Id: Unique identifier of the session Reason: Error message
3007	Update session review status succeeded added in release 18.8	Installation: Name of the installation Session Id: Unique identifier of the session Review Status: Name of the review status
3008	Update session review status failed added in release 18.8	Installation: Name of the installation Session Id: Unique identifier of the session Review Status: Name of the review status Reason: Error message
3009	Replay session succeeded Added in release 19.6	Installation: Name of the installation Session Id: Unique identifier of the session User: User of the session Machine: Machine of the session
3010	Replay session failed Added in release 19.6	Installation: Name of the installation Session Id: Unique identifier of the session Reason: Error message

3011	Delete audit trail events succeeded Added in release 19.9	SearchFilter: Search Filter
3012	Delete audit trail events failed Added in release 19.9	SearchFilter: Search Filter Reason: Error Message
3013	Delete session succeeded Added in release 2020.1	Session Id: Unique identifier of the session Username: Name of the user whose session was recorded Machinename: Name of the machine where the session was recorded
3014	Delete session failed Added in release 2020.1	Session Id: Unique identifier of the session Username: Name of the user whose session was recorded Machinename: Name of the machine where the session was recorded Reason: error message

Audit Manager

Audit Manager is a Microsoft management console (MMC) that you can use to configure and manage the deployment of audit components, such as audit stores and audit store databases, audit roles, collectors, and agents. Audit Manager is available with Server Suite. Audit events generated by Audit Manager primarily involve the installation and configuration of auditing components such as management databases, audit stores, and audit store databases, and changes to audit role and user permissions.

Audit Analyzer Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 3001. This log sample documents a session being deleted. The change was made by user=administrator@acme.vms on April 20, 2016 at 05:51:01.

```
04/20/2016 05:51:01 PM LogName=Application
SourceName=Centrify AuditTrail V2 EventCode=3001
EventType=4 Type=Information ComputerName=
member.acme.vms User=NOT_TRANSLATED Sid=S-1-
5-21-3883016548-1611565816-1967702834-500 SidType=0
TaskCategory=%1 OpCode=Info RecordNumber=60622
Keywords=Classic Message=Product: Centrify Suite Category:
Audit Analyzer Event name: Delete session Message: 1 out
of 1 selected sessions are successfully deleted. Apr 20
17:51:00 member.acme.vms mmc[4064]: INFO
AUDIT_TRAIL|Centrify Suite|Audit Analyzer|1.0|1|Delete
session|5|user=administrator@acme.vms
userSid=S-1-5-21-3883016548-1611565816-1967702834-500
sessionId=11 CentrifyEventID=3001 DAInst=
AuditingInstallation DASessID=c72252aa-e616-44ff-a5f6-
d3f53f09bb67 sessions_deleted=1 sessions_selected=1
```

Audit Manager Audit Events

Audit Manager Audit Events

12200	Video capture status updatedv	installation: audit and monitoring service Installation VideoCaptureStatus: video capture status
12201	Create new installation succeededv	installation: Name of the installation
12202	Create new installation failedv	installation: Name of the installation reason: Error message
12203	Installation update succeededv	installation: Name of the installation Installation Property: Name of the updated installation property Installation Property Value: Value of the updated installation property Operation: Type of operation (Set or Add or Remove)
12204	Installation update failedv	installation: Name of the installation Installation Property: Name of the updated installation property Installation Property Value: Value of the updated installation property Operation: Type of operation (Set or Add or Remove) reason: Error message
12205	Installation permissions update succeededv	installation: Name of the installation User/Group: Name of the user or group Permissions: Permissions assigned to the user or group
12206	Installation permissions update failed	installation: Name of the installation User/Group: Name of the user or group Permissions: Permissions assigned to the user or group reason: Error message
12207	Remove installation succeeded	installation: Name of the installation

12208	Remove installation failed	installation: Name of the installation reason: Error message
12251	Audit options updated	installation: audit and monitoring service Installation DisableSelfReview: Disable reviewing own sessions DisableSelfDelete: Disable deleting own sessions
12209	Add Management Database succeeded	installation: Name of the installation Management Database: Name of the Management Database
12210	Add Management Database failed	installation: Name of the installation Management Database: Name of the Management Database reason: Error message
12211	Management Database update succeeded	installation: Name of the installation Management Database: Name of the Management Database Management Database Property: Name of the updated Management Database property Management Database Property Value: Value of the updated Management Database property Operation: Type of operation (Set or Add or Remove)
12212	Management Database update failed	installation: Name of the installation Management Database: Name of the Management Database Management Database Property: Name of the updated Management Database property Management Database Property Value: Value of the updated Management Database property Operation: Type of operation (Set or Add or Remove) reason: Error message
12213	Management Database permissions update succeeded	installation: Name of the installation Management Database: Name of the Management Database User/Group: Name of the user or group Permissions: Permissions assigned to the user or group
12214	Management Database permissions update failed	installation: Name of the installation Management Database: Name of the Management Database User/Group: Name of the user or group Permissions: Permissions assigned to the user or group reason: Error message
12215	Remove Management Database succeeded	installation: Name of the installation Management Database: Name of the Management Database
12216	Remove Management Database failed	installation: Name of the installation Management Database: Name of the Management Database reason: Error message
12217	Add Audit Store succeeded	installation: Name of the installation Audit Store: Name of the Audit Store
12218	Add Audit Store failed	installation: Name of the installation Audit Store: Name of the Audit Store reason: Error message
12219	Audit Store update succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Property: Name of the updated Audit Store property Audit Store Property Value: Value of the updated Audit Store property Operation: Type of operation (Set or Add or Remove)
12220	Audit Store update failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Property: Name of the updated Audit Store property Audit Store Property Value: Value of the updated Audit Store property Operation: Type of operation (Set or Add or Remove) reason: Error message

12221	Audit Store permissions update succeeded	installation: Name of the installation Audit Store: Name of the Audit Store User/Group: Name of the user or group Permissions: Permissions assigned to the user or group
12222	Audit Store permissions update failed	installation: Name of the installation Audit Store: Name of the Audit Store User/Group: Name of the user or group Permissions: Permissions assigned to the user or group reason: Error message
12223	Remove Audit Store succeeded	installation: Name of the installation Audit Store: Name of the Audit Store
12224	Remove Audit Store failed	installation: Name of the installation Audit Store: Name of the Audit Store reason: Error message
12225	Add Audit Store Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database
12226	Add Audit Store Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database reason: Error message
12227	Attach Audit Store Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database
12228	Attach Audit Store Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database reason: Error message
12229	Attach audit and monitoring service Version 1 Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the audit and monitoring service Version 1 Database
12230	Attach audit and monitoring service Version 1 Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the audit and monitoring service Version 1 Database reason: Error message
12231	Set Active Audit Store Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database
12232	Set Active Audit Store Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database reason: Error message
12233	Audit Store Database update succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database Audit Store Database Property: Name of the updated Audit Store Database property Audit Store Database Property Value: Value of the updated Audit Store Database property Operation: Type of operation (Set or Add or Remove)
12234	Audit Store Database update failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database Audit Store Database Property: Name of the updated Audit Store Database property Audit Store Database Property Value: Value of the updated Audit Store Database property Operation: Type of operation (Set or Add or Remove) reason: Error message

12235	Detach Audit Store Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database
12236	Detach Audit Store Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database reason: Error message
12237	Delete Audit Store Database succeeded	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database
12238	Delete Audit Store Database failed	installation: Name of the installation Audit Store: Name of the Audit Store Audit Store Database: Name of the Audit Store Database reason: Error message
12239	Add Audit Role succeeded	installation: Name of the installation Audit Role: Name of the Audit Role
12240	Add Audit Role failed	installation: Name of the installation Audit Role: Name of the Audit Role reason: Error message
12241	Audit Role update succeeded	installation: Name of the installation Audit Role: Name of the Audit Role Audit Role Property: Name of the updated Audit Role property Audit Role Property Value: Value of the updated Audit Role property Operation: Type of operation (Set or Add or Remove)
12242	Audit Role update failed	installation: Name of the installation Audit Role: Name of the Audit Role Audit Role Property: Name of the updated Audit Role property Audit Role Property Value: Value of the updated Audit Role property Operation: Type of operation (Set or Add or Remove) reason: Error message
12243	Audit Role permissions update succeeded	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group Permissions: Permissions assigned to the user or group
12244	Audit Role permissions update failed	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group Permissions: Permissions assigned to the user or group reason: Error message
12245	Audit Role assign member succeeded	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group
12246	Audit Role assign member failed	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group reason: Error message
12247	Audit Role remove member succeeded	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group
12248	Audit Role remove member failed	installation: Name of the installation Audit Role: Name of the Audit Role User/Group: Name of the user or group reason: Error message
12249	Delete Audit Role succeeded	installation: Name of the installation Audit Role: Name of the Audit Role
12250	Delete Audit Role failed	installation: Name of the installation Audit Role: Name of the Audit Role reason: Error message

Centrify Commands (UNIX Commands)

Audit events in the Centrify Commands category are focused on capturing command line activity. Audit events are recorded when users or administrators run command line programs to enable or disable auditing, join or leave a domain, query Active Directory for user or group information, change their password configuration settings or license mode, or perform other operations.

Centrify Command Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 18000. This log sample documents auditing being enabled. The change was made by user=root on April 5 at 11:37:28.

```
Apr 5 11:37:28 engcen6 adclient[1749]: INFO AUDIT_
TRAIL|Centrify Suite|Centrify Commands|1.0|0|Auditing
enabled|5|user=root pid=14874 utc=1459836448489
CentrifyEventID=18000 DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=GRANTED service=NSS
```

Centrify Commands Audit Events

Centrify Commands Audit Events

18000	Auditing enabled	service: service
18001	Auditing not enabled	service: service reason: error message
18100	Auditing disabled	service: service
18101	Auditing not disabled	service: service reason: error message
18200	The user login to the system successfully	service: service tty: tty
18300	Desktop auditing enabled Added in Release 2020	
18301	Desktop auditing not enabled Added in Release 2020	reason: error message
18400	Desktop auditing disabled Added in Release 2020	
18401	Desktop auditing not disabled Added in Release 2020	reason: error message
18500	Session auditing started Added in Release 2020	
18501	Session auditing ended Added in Release 2020	
20100	Joined domain	parameters: parameters zone: zone name domain: domain computer: computer name runas: username@domain
20101	Join failed	parameters: parameters zone: zone name domain: domain computer: computer name runas: username@domain

		reason: error message
20200	Left domain	parameters: parameters
20201	Leaving domain failed	parameters: parameters reason: error message
20300	Query as root was successful	parameters: parameters
20301	Query was successful	parameters: parameters
20302	Query request failed	parameters: parameters reason: error message
20400	Password changed	parameters: parameters unixUser: user name
20401	Password change failed	parameters: parameters unixUser: user name reason: error message
20500	Configuration settings (Centrify.conf) reloaded	parameters: parameters
20501	Configuration settings (Centrify.conf) failed to reload	parameters: parameters reason: error message
20600	Local cache flushed	parameters: parameters
20601	Cache flush failed	parameters: parameters reason: error message
20650	Object refreshed	parameters: parameters
20651	Object refresh failed	parameters: parameters reason: error message
20800	License modes changed	parameters: parameters
20801	License modes change failed	parameters: parameters reason: error message
20900	Advanced monitoring enabled	service: service
20901	Advanced monitoring not enabled	service: service reason: error message
20910	Advanced monitoring disabled	service: service
20911	Advanced monitoring not disabled	service: service reason: error message
21100	Changing web proxy configuration succeeded added in release 18.8	parameters: parameters
21101	Changing web proxy configuration failed added in release 18.8	parameters: parameters reason: error message

21200	Editing Kerberos keytab file succeeded	parameters: parameters
21201	Editing Kerberos keytab file failed	parameters: parameters reason: error message

Centrify Configuration

Centrify hierarchical zones are used to enable information about non-Windows computers, user profiles, access rights, and roles to be stored in Active Directory. Hierarchical zones can be used to segregate and perform privilege management on both UNIX/Linux and Windows systems. These configuration audit events focus on zones, computers, groups, users, rights, and roles.

Centrify Configuration Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 36101. This log sample documents a user giving zone administrative tasks to another user. The change was made by user=dwirth@acme.vms on April 19, 2016 at 03:01:04.

```
04/19/2016 03:01:04 PM LogName=Application
SourceName=CentrifyAuditTrail V2 EventCode=36101
EventType=4 Type=Information
ComputerName=member.acme.vms
User=NOT_TRANSLATED Sid=S-1-5-21-3883016548-1611565816-
1967702834-1107 SidType=0 TaskCategory=%1 OpCode=Info RecordNumber=59436
Keywords=Classic Message=Product:
Centrify Suite Category: Centrify Configuration Event
name: Zone administrative tasks delegated Message:
"dwirth@acme.vms" (running as "dwirth@acme")
delegated "acmepankaj" to perform "Change zone
properties" on "acme.vms/acmese/Zones/zone-14".
Apr 19 15:01:04 member mmc[5792]: INFO AUDIT_TRAIL|Centrify
Suite|CentrifyvConfiguration|1.0|101|Zone
tasks delegated|5|user=dwirth@acme.vms userSid=
S-1-5-21-3883016548-1611565816-1967702834-1107 sessionId=3
CentrifyEventID=36101 DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67 pid=5792
user=dwirth@acme.vms runas=dwirth@acme type=AD
status=SUCCESS trustee=acmepankaj task=Change zone
properties zone=acme.vms/acmese/Zones/zone-14
```

Centrify Configuration Audit Events

Centrify Configuration Audit Events

36101	Zone administrative tasks delegated	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded trustee: username@domain task: delegation task name zone: zone name
36102	Delegation of zone administrative tasks failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed trustee: username@domain task: delegation task name zone: zone name reason: failure reason
36103	Computer administrative tasks delegated	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded trustee: username@domain task: delegation task name zone: zone name computer: computer name
36104	Delegation of computer administrative tasks failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed trustee: username@domain task: delegation task name zone: zone name computer: computer name reason: error message
36105	Computer role administrative tasks delegated	PID: process id user: username@domain RunAs: username@domain type: user type status: trustee: username@domain task: delegation task name zone: zone name computerRole: computer role name
36106	Delegation of computer role administrative tasks failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed trustee: username@domain task: delegation task name zone: zone name computerRole: computer role name reason: error message
36201	Zone created	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name

36202	Zone creation failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed zone: zone name reason: error message
36203	Zone deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name
36204	Zone deletion failed	status: failed PID: process id user: username@domain RunAs: username@domain type: user type zone: zone name reason: error message
36205	Zone modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name
36206	Zone update failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed zone: zone name reason: error message
36301	User added to a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
36302	Add user to a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message
36303	User deleted from a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
36304	Delete user from a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message
36305	User profile modified in a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
36306	Modify user in a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message
36307	User added to a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded : unixname computer: computer hostname zone: zone name
36308	Add user to a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
36309	User deleted from computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname computer: computer hostname zone: zone name
36310	Delete user from a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
36311	User profile modified on a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname computer: computer hostname zone: zone name
36312	Modify user on a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
36401	Group added to a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name

36402	Add group to a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
36403	Group deleted from a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name
36404	Delete group from a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
36405	Group profile modified in a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name
36406	Modify group in a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
36407	Group added to a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
36408	Add group to a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message
36409	Group deleted from a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
36410	Delete group from a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message
36411	Group profile modified on a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
36412	Modify group for a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message
36501	Computer added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computer: hostname zone: zone name
36502	Add computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: hostname zone: zone name reason: error message
36503	Computer deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computer: hostname zone: zone name
36504	Delete computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: hostname zone: zone name reason: error message
36505	Computer modified	PID: process id user: username@domain RunAs: username@domain type: user type status: computer: hostname zone: zone name
36506	Modify computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: hostname zone: zone name reason: error message
36601	PAM access right added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded pam: pam name zone: zone name

36602	Add PAM right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed pam: pam name zone: zone name reason: error message
36603	PAM right deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded pam: pam name zone: zone name
36604	Delete PAM right failed	PID: process id user: username@domain RunAS: username@domain type: user type status: failed pam: pam name zone: zone name reason: error message
36605	PAM right modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded pam: pam name zone: zone name
36606	Modify PAM right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed pam: pam name zone: zone name reason: error message
37201	Desktop right added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded desktop: desktop right name zone: zone name
37202	Add Desktop Right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed desktop: desktop right name zone: zone name reason: error message
37203	Desktop right deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded desktop: desktop right name zone: zone name
37204	Delete desktop right failed	PID: process id user: username@domain RunAS: username@domain type: user type status: failed desktop: desktop right name zone: zone name reason: error message
37205	desktop right modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded desktop: desktop right name zone: zone name
37206	Modify desktop right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed desktop: desktop right name zone: zone name reason: error message
37301	Network right added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded network: network right name zone: zone name
37302	Add network right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed network: network right name zone: zone name reason: error message
37303	network right deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded network: network right name zone: zone name
37304	Delete network right failed	PID: process id user: username@domain RunAS: username@domain type: user type status: failed network: network right name zone: zone name reason: error message
37305	Network right modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded network: network right name zone: zone name
37306	Modify network right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed network: network right name zone: zone name reason: error message
37401	Application right added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded application: application right name zone: zone name
37402	Add application right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed application: application right name zone: zone name reason: error message

		application right name zone: zone name reason: error message
37403	Application right deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded application: application right name zone: zone name
37404	Delete application right failed	PID: process id user: username@domain RunAS: username@domain type: user type status: failed application: application right name zone: zone name reason: error message
37405	Application right modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded application: application right name zone: zone name
37406	Modify application right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed application: application right name zone: zone name reason: error message
36701	UNIX command right added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded dzcmd: dzcmd zone: zone name
36702	Add command right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed dzcmd: dzcmd zone: zone name reason: error message
36703	UNIX command right deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded dzcmd: dzcmd zone: zone name
36704	Delete command right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed dzcmd: dzcmd zone: zone name reason: error message
36705	UNIX command right modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded dzcmd: dzcmd zone: zone name
36706	Modify command right failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed dzcmd: dzcmd zone: zone name reason: error message
36801	Role added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded role: role name zone: zone name
36802	Add role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed role: role name zone: zone name reason: error message
36803	Role deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded role: role name zone: zone name
36804	Delete role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed role: role name zone: zone name reason: error message
36805	Role modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded role: role name zone: zone name
36806	Modify role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed role: role name zone: zone name reason: error message
36807	Add right to role was successful	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded right: right name role: role name
36808	Add right to role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed right: right name role: role name reason: error message

36809	Delete right from role was successful	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded right: right name role: role name
36810	Delete right from role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed right: right name role: role name reason: error message
36901	Role assignment added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name role: role name trustee: username@domain
36902	Role assignment failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed zone: zone name role: role name trustee: username@domain reason: error message
36903	Role assignment removed	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name role: role name trustee: username@domain
36904	Delete role assignment failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed zone: zone name role: role name trustee: username@domain reason: error message
36905	Role assignment modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded zone: zone name role: role name trustee: username@domain
36906	Modify role assignment failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed zone: zone name role: role name trustee: username@domain reason: error message
36907	Role assignment added to a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computer: computer zone: zone name role: role name trustee: username@domain
36908	Add role assignment to computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: computer hostname zone: zone name role: role name trustee: username@domain reason: error message
36909	Role assignment deleted from a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: computer: computer hostname zone: zone name role: role name trustee: username@domain
36910	Delete role assignment from computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: computer hostname zone: zone canonical role: role name trustee: username@domain reason: error message
36911	Role assignment modified for a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computer: computer hostname zone: zone canonical role: role name trustee: username@domain
36912	Modify role assignment for a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computer: computer hostname zone: zone canonical role: role name trustee: username@domain reason: error message
36913	Role assignment added to a computer role	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computerRole: computer role zone: zone name role: role name trustee: username@domain
36914	Role assignment for a computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role name zone: zone name role: role name trustee: username@domain reason: error message
36915	Role assignment deleted from a computer role	PID: process id user: username@domain RunAs: username@domain type: user type status: computerRole: computer role name zone: zone name role: role name trustee: username@domain
36916	Delete role assignment from a computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role name zone: zone canonical role: role name trustee: username@domain reason: error message

36917	Role assignment modified for a computer role	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computerRole: computer role name zone: zone canonical role: role name trustee: username@domain
36918	Modify role assignment in a computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role name zone: zone canonical role: role name trustee: username@domain reason: error message
37001	Computer role added	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computerRole: computer role name zone: zone name
37002	Add computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role name zone: zone name reason: error message
37003	Computer role deleted	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computerRole: computer role name zone: zone name
37004	Delete computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role name zone: zone name reason: error
37005	Computer role modified	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded computerRole: computer role name zone: zone name
37006	Modify computer role failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed computerRole: computer role zone: zone name reason: error message
37101	User added to a group	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded member: username group: group name
37102	Add user to a group failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed member: username group: group name reason: error message
37103	Password reset	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded account: username
37104	Reset password failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed account: username reason: error message
37501	user added to a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
37502	Add local user to a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message
37503	Local user deleted from a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
37504	Delete local user from a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message
37505	Local user profile modified in a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname zone: zone name
37506	Modify local user in a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname zone: zone name reason: error message

37511	Local user added to a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname computer: computer hostname zone: zone name
37512	Add local user to a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
37513	Local user deleted from computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname computer: computer hostname zone: zone name
37514	Delete local user from a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
37515	Local user profile modified on a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded ZoneUser: unixname computer: computer hostname zone: zone name
37516	Modify local user on a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed ZoneUser: unixname computer: computer hostname zone: zone name reason: error message
37521	Local group added to a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name
37522	Add local group to a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
37523	Local group deleted from a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name
37524	Delete local group from a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
37525	Local group profile modified in a zone	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name zone: zone name
37526	Modify local group in a zone failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name zone: zone name reason: error message
37531	Local group added to a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
37532	Add local group to a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message
37533	Local group deleted from a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
37534	Delete local group from a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message
37535	Local group profile modified on a computer	PID: process id user: username@domain RunAs: username@domain type: user type status: succeeded group: group name computer: computer hostname zone: zone name
37536	Modify local group for a computer failed	PID: process id user: username@domain RunAs: username@domain type: user type status: failed group: group name computer: computer hostname zone: zone name reason: error message

37601	Local Windows user added to a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name
37602	Add local Windows user to a zone failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name reason: error message
37603	Local Windows user deleted from a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name
37604	Delete local Windows user from a zone failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name reason: error message
37605	Local Windows user modified in a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name
37606	Modify local Windows user in a zone failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name zone: zone name reason: error message
37611	Local Windows user added to a computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name
37612	Add local Windows user to a computer failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name reason: error message
37613	Local Windows user deleted from computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name
37614	Delete local Windows user from a computer failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name reason: error message
37615	Local Windows user modified on a computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name
37616	Modify local Windows user on a computer failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed ZoneUser: local Windows user name computer: computer hostname zone: zone name reason: error message
37621	Local Windows group added to a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name zone: zone name
37622	Add local Windows group to a zone failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name zone: zone name reason: error message
37623	Local Windows group deleted from a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name zone: zone name
37624	Local Windows group modified in a zone added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name zone: zone name
37626	Modify local Windows group in a zone failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name zone: zone name reason: error message

37631	Local Windows group added to a computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name
37632	Add local Windows group to a computer failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name reason: error message
37633	Local Windows group deleted from a computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name
37634	Delete local Windows group from a computer failed added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name reason: error message
37635	Local Windows group modified on a computer added in Release 2020	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name
37636	Modify local Windows group for a computer failed added in Release	PID: process ID user: username@domain RunAs: username@domain type: user type status: succeeded or failed group: group name computer: computer hostname zone: zone name reason: error message

Centrify sshd

Centrify sshd is Centrify's enhanced version of OpenSSH. This software program uses the secure shell protocol to connect to a remote computer. Centrify sshd audit events identify DZ SSH rights and SSHD activities.

Centrify sshd Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 27000. This log sample documents the rights granted to the DZ SSH shell client. The change was made by user=dwirth(type:ad,dwirth@acme.vms) on April 4 at 01:04:15.

```
Apr 4 21:04:15 engcen6 adclient[1749]: INFO
AUDIT_TRAILICentrify SuitelCentrify sshdl1.0l0lDZ SSH right
grantedl5luser=dwirth(type:ad,dwirth@acme.vms) pid=7461
utc=1459784055474 CentrifyEventID=27000
DAInst=AuditingInstallation DASessID=c72252aa-e616-
44ff-a5f6-d3f53f09bb67 status=GRANTED
service=dzssh-shell client=192.168.81.11
```

Centrify sshd Audit Events

Centrify sshd Audit Events

27000	DZ SSH right granted	service: service client: client
27001	DZ SSH right denied	service: service client: client reason: error message
27100-Deprecated	SSHD granted This event has been deprecated. Use Centrify Event Id 27104 introduced in release 2017.3 instead.	service: service tty: tty authMechanism: authentication type client: client sshRights: ssh rights command: command
27101-Deprecated	SSHD denied This event has been deprecated. Use Centrify Event Id 27105 introduced in release 2017.3 instead.	service: service tty: tty authMechanism: authentication type client: client reason: error message
27102	SSHD connection close successfully	service: service tty: tty authMechanism: authentication type client: client reason: error message
27104	SSHD granted added in release 2017.3	service: service tty: tty authMechanism: authentication type client: client sshRights: ssh rights command: command MfaRequired: whether user was required to do MFA EntityName: Entity Name
27105	SSHD denied added in release 2017.3	service: service tty: tty authMechanism: authentication type client: client reason: error message MfaRequired: whether user was required to do MFA EntityName: Entity Name
27200	SCP succeeded added in release 18.8	dataFlowType: send a file/directory to remote machine or receive a file/directory from remote machine fileType: file or directory pathname: the full path name of file or directory
27201	SCP failed added in release 18.8	dataFlowType: send a file/directory to remote machine or receive a file/directory from remote machine fileType: file or directory pathname: the full path name of file or directory reason: Error message
27300	SFTP command execution succeeded added in release 18.8	operation: SFTP command arguments: the arguments of SFTP command
27301	SFTP command execution failed added in release 18.8	

) [tags]: # (server suite) [priority]: # (12)

Command (Audited and Successfully Executed Commands)

Command audit events are recorded when Centrify UNIX command-line programs are used on Centrify-managed computers. Centrify UNIX command audit events focus on the execution success or failure of the audited command.

Command Audit Event Log Sample

```
Nov 26 00:32:01 Eason adclient[31118]: INFO
AUDIT_TRAILICentrify SuitelCommandl1.0l100
lAudited command is executedl5luser=
pid=31937 utc=1416979921469 CentrifyEventID=48100
DAInst=AuditingInstallation DASessID=c72252aa-e616
-44ff-a5f6-d3f53f09bb67 status=SUCCESS
command=/bin/ls -l data.txt
```

Command Audit Events

Event Source Category: Command

48100	Audited command is executed	command: command
48101	Audited command fails to be executed	command: command reason: error message

Centrify Audit & Monitoring Service Advanced Monitoring

If you have enabled Centrify Audit & Monitoring Service for advanced monitoring, you can generate data for three additional auditing reports, as follows:

- Monitored execution report: This report shows the monitored commands being executed on the audited machines—including information on commands that are run individually or as part of scripts.
- Detailed execution report: This report shows all of the commands being executed on the audited machines—including commands that are run as part of scripts or other commands.
- File monitor report: This report shows the sensitive files being modified by users on the audited machines.

Advanced Monitoring Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 57300. This log sample documents a session where a user attempted to modify a monitored file. The change was made by root@al_rhel6_2.altest.acme.com on November 2, 2016 at 06:09:01.

```
Nov 2 06:09:01 al_rhel6_2 adclient[27002]: INFO
AUDIT_TRAILICentrify SuiteIDirectAudit Advanced
MonitoringI1.0I300IMonitored file modification
attemptedI5Iuser=<no_login_user> pid=32393
utc=1478092141432 CentrifyEventID=57300
DAInst=AuditingInstallation DASessID=c72252aa-
e616-44ff-a5f6-d3f53f09bb67 status=SUCCESS
syscall=unlink status=0 timestamp=1478092141.432000
auid=<no_login_user> uid=root@al_rhel6_2.altest.
acme.com processid=32393 ppid=32392 gid=root
euid=root@al_rhel6_2.altest.acme.com cwd=/ accessType=2
command=/usr/bin/python argc=-1 args=/etc/pki/nssdb/
/etc/pki/nssdb/cert9.db-journal
```

Centrify Audit & Monitoring Service Advanced Monitoring Audit Events

Audit and Monitoring Service Advanced Monitoring Audit Events

57200	Monitored program is executed	syscall: system call exitcode: exit code timestamp: timestamp auid: login user uid: user procid: process id ppid: parent process id gid: group euid: effective user cwd: current working directory cmd: command argc: no of arguments args: arguments
57201	Monitored program failed to execute	syscall: system call exitcode: exit code timestamp: timestamp auid: login user uid: user procid: process id ppid: parent process id gid: group euid: effective user cwd: current working directory cmd: command argc: no of arguments args: arguments
57300	Monitored file modification attempted	syscall: system call exitcode: exit code timestamp: timestamp auid: login user uid: user procid: process id ppid: parent process id gid: group euid: effective user cwd: current working directory accType: access Type cmd: command argc: no of arguments args: arguments
57301	Monitored file modification attempt failed	syscall: system call exitcode: exit code timestamp: timestamp auid: login user uid: user procid: process id ppid: parent process id gid: group euid: effective user cwd: current working directory accType: access Type cmd: command argc: no of arguments args: arguments
57400	Command execution is started	syscall: syscall exitcode: exit code timestamp: timestamp auid: auid uid: uid pid: pid ppid: ppid gid: gid euid: euid cwd: current working directory command: command argc: no of arguments args: arguments
57401	Command execution fails to start	syscall: syscall exitcode: exit code timestamp: timestamp auid: auid uid: uid pid: pid ppid: ppid gid: gid euid: euid cwd: current working directory command: command argc: no of arguments args: arguments

Centrify Audit & Monitoring Service System Management

The auditing module's detailed, real-time auditing of privileged user sessions on Windows, UNIX, and Linux systems provides a full accounting of user activity and system access. Centrify Audit & Monitoring Service System Management is available with Centrify Audit & Monitoring Service. The audit and monitoring service audit events focus on collector service, collector settings, and agent settings.

Centrify Audit & Monitoring Service System Management audit event log sample

The following is a sample of an audit event log for Centrify Audit Event ID 42251. This log sample documents the successful start of the collector service on computer 'MEMBER'. The change was made by user=system@nt authority on April 05, 2016 at 14:59:56.

```
04/05/2016 03:00:01 PM LogName=Application SourceName=
 Centrify AuditTrail V2 EventCode=42251 EventType=4
 Type=Information ComputerName=member.acme.vms
 User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0
 TaskCategory=%1 OpCode=Info RecordNumber=51722
Keywords=Classic Message=Product: Centrify Suite Category:
 DirectAudit System Management Event name: Start collector
 service succeeded Message: Collector service was started
 successfully on computer 'MEMBER'. Apr 05 14:59:56
 member.acme.vms collector[1344]: INFO AUDIT_TRAIL|
 Centrify Suite|DirectAudit System Management|1.0|251|Start
 collector service succeeded|5|user=system@nt authority
 userSid=S-1-5-18 sessionId=0 centrifyEventID=42251
 DAInst=AuditingInstallation DASessID=c72252aa-e616-
 44ff-a5f6-d3f53f09bb67 installation=DefaultInstallation
collector=MEMBER
```

Centrify Audit & Monitoring Service System Management audit events

Audit and Monitoring Service System Management Audit Events

42251	Start collector service succeeded	installation: Name of the installation Collector: Name of the collector computer
42252	Start collector service failed	installation: Name of the installation Collector: Name of the collector computer reason: Error message
42253	Stop collector service succeeded	installation: Name of the installation Collector: Name of the collector computer
42254	Stop collector service failed	installation: Name of the installation Collector: Name of the collector computer reason: Error message
42255	Collector settings update succeeded	installation: Name of the installation Collector: Name of the collector computer Collector setting: Name of the updated collector setting Collector setting value: Value of the updated collector setting
42256	Collector settings update failed	installation: Name of the installation Collector: Name of the collector computer Collector setting: Name of the updated collector setting Collector setting value: Value of the updated collector setting reason: Error message
42257	Start agent service succeeded	installation: Name of the installation Audited system: Name of the audited system
42258	Start agent service failed	installation: Name of the installation Audited System: Name of the audited system reason: Error message
42259	Stop agent service succeeded	installation: Name of the installation Audited system: Name of the audited system
42260	Stop agent service failed	installation: Name of the installation Audited system: Name of the audited system reason: Error message
42261	Agent settings update succeeded	installation: Name of the installation Audited system: Name of the audited system Agent setting: Name of the updated agent setting Agent setting value: Value of the updated agent setting

42262	Agent settings update failed	installation: Name of the installation Audited system: Name of the audited system Agent setting: Name of the updated agent setting Agent setting value: Value of the updated agent setting reason: Error message
42263	Start audit management service succeeded added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer
42264	Start audit management service failed added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer reason: Error message
42265	Stop audit management service succeeded added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer
42266	Stop audit management service failed added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer reason: Error message
42267	Started the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name
42268	Failed to start the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name reason: Error message
42269	Stopped the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name
42270	Failed to stop the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name reason: Error message
42271	Restarted the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name
42272	Failed to restart the collector service added in release 18.11	installation: Name of the installation Collector: Name of the collector computer User: User name reason: Error message
42273	Started the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name
42274	Failed to start the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name reason: Error message
42275	Stopped the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name
42276	Failed to stop the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name reason: Error message
42277	Restarted the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name
42278 Good	Failed to restart the audit management service added in release 18.11	installation: Name of the installation Audit Management: Name of the audit management computer User: User name reason: Error message

Centrify Authentication Service UNIX Agent

The Centrify Authentication Service UNIX Agent audit events are focused on the success or failure of starting and stopping the Centrify Agent: **adclient**.

Centrify Authentication Service UNIX Agent Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 17000. This log sample documents the successful start of the Centrify Agent: adclient. The change was made by user=root on April 05 at 06:46:43.

```
Apr 5 06:46:43 newcentos adclient[1837]: INFO AUDIT_
TRAIL|Centrify Suite|DirectControl UNIX Agent|1.0|2000
Centrify Agent (adclient) started|5|user=root pid=1837
utc=1459856803582 CentrifyEventID=17000
DAInst=AuditingInstallation DASessID=c72252aa-
e616-44ff-a5f6-d3f53f09bb67 status=SUCCESS service=adclient
```

Centrify Authentication Service UNIX Agent Audit Events

Authentication Service UNIX Agent Audit Events

17000	Centrify Agent (adclient) started	
17001	Centrify Agent (adclient) failed to start	reason: error message
17002	Centrify Agent (adclient) stopped	
17003	Centrify Agent (adclient) failed to stop	reason: error message

Centrify Audit & Monitoring Service – Windows

Centrify Audit & Monitoring Service collects login success audit data from Windows computers. The Centrify Audit & Monitoring Service audit event focuses on login success.

Centrify Audit & Monitoring Service – Windows Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 9001. This log sample documents a successful login. The change was made by user=administrator@acme.test on January 06 at 15:53:10.

```
Jan 06 15:53:10 s2k8r2p1v1.acme.test wdad[1128]:
INFO AUDIT_TRAILICentrify SuitelDirectAudit -
Windowsl1.0l1llogin successl5luser=administrator
@acme.test userSid=S-1-5-21-1986235188-3370598863-
2160698129-500 sessionId=1 CentrifyEventID=9001
DAInst=AuditingInstallation DASessID=c72252aa-
e616-44ff-a5f6-d3f53f09bb67
```

Centrify Audit & Monitoring Service - Windows Audit Events

Audit and Monitoring Service - Windows Audit Events

9001	login success	
9002	logoff success	
9003	Enable Centrify Auditing and Monitoring Service succeeded added in release 2017.3	InstallationName: Installation Name
9004	Disable Centrify Auditing and Monitoring Service succeeded added in release 2017.3	InstallationName: Installation Name
9005	Enable Centrify Auditing and Monitoring Service failed added in release 2017.3	InstallationName: Installation Name Reason: Reason for failure
9006	Disable Centrify Auditing and Monitoring Service failed added in release 2017.3	InstallationName: Installation Name Reason: Reason for failure
9007	Session auditing started added in Release 2020	
9008	Session auditing ended added in Release 2020	

Centrify Privilege Elevation Service – Windows

Centrify Privilege Elevation Service for Windows provides role-based access control for Windows desktops and applications, and to remote Windows servers. Centrify Privilege Elevation Service for Windows audit events focus on successful and failed local console and remote log in attempts, administrative activity using desktop or application privileges, network access to remote servers, changes to the zone information for Windows computers and changes to role information for Windows users.

Centrify Privilege Elevation Service Windows Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 6029. This log sample documents a user with local and network role privileges launching a .msc file.

Log Name: Application
Source: Centrify AuditTrail V2
Date: 9/19/2019 2:05:17 PM
Event ID: 6029
Task Category: None
Level: Information
Keywords: Classic
User: bob@acme.vms
Computer: member.acme.vms
Description:
Product: Centrify Suite
Category: DirectAuthorize - Windows
Event name: Run with privilege success
Message: User launched 'C:Program FilesCentrifyAccess
ManagerCentrifyDC.msc' on
desktop 'Default' using local role 'ROLE_SYSTEM_Archt/Global'
and network roles 'ROLE_SYSTEM_Archt/Global'.
Sep 19 14:05:17 member.acme.vms dzagent[1348]:
INFO AUDIT_TRAIL|Centrify Suite|DirectAuthorize - Windows|1.0|29|Run with
privilege
success|5|bob@acme.vms
userSid=S-1-5-21-569763308-1211465464-1224152175-3219
sessionId=3 CentrifyEventID=6029
DAInst=AuditingInstallation DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
role=ROLE_SYSTEM_Archt/Global
effectivesid=S-1-5-21-569763308-1211465464-1224152175-3219
effectivegroupsids=S-1-5-32-544
logonguid=ad7b6538-e2a4-4304-ab6e-86c5b0dabfaf
desktopguid=1e09a3dd-276f-4629-bb27-e215dfe0a0c8
command=C:Program FilesCentrifyAccessManagerCentrifyDC.msc
passwordprompted=False desktopname=Default
networkroles=ROLE_SYSTEM_Archt/Global
entityname=acme.vms mfarequired=False

Centrify Privilege Elevation Service - Windows Audit Events

Privilege elevation service - Windows Audit Events

6001-Deprecated	Console login success This event has been deprecated. Use Centrify Event Id 6031 introduced in release 2017.2 instead.	Role: role DesktopGuid: desktop GUID
6002-Deprecated	Console login failure This event has been deprecated. Use Centrify Event Id 6032 introduced in release 2017.2 instead.	
6003-Deprecated	Remote login success This event has been deprecated. Use Centrify Event Id 6033 introduced in release 2017.2	Role: role DesktopGuid: desktop GUID

	instead.	
6004-Deprecated	Remote login failure This event has been deprecated. Use Centrify Event Id 6034 introduced in release 2017.2 instead.	
6005-Deprecated	Run with privilege success This event has been deprecated. Use Centrify Event Id 6029 introduced in release 2017.2 instead.	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID Command: command
6006-Deprecated	Run with privilege failure This event has been deprecated. Use Centrify Event Id 6030 introduced in release 2017.2 instead.	Role: local role DesktopGuid: desktop GUID Command: command
6007-Deprecated	Create desktop success This event has been deprecated. Use Centrify Event Id 6035 introduced in release 2017.2 instead.	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID
6008-Deprecated	Create desktop failure This event has been deprecated. Use Centrify Event Id 6036 introduced in release 2017.2 instead.	Role: local role
6009-Deprecated	Network access success This event has been deprecated. Use Centrify Event Id 6039 introduced in release 2017.2 instead.	Role: role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID
6010-Deprecated	Console logon failure This event has been deprecated. Use Centrify Event Id 6032 introduced in release 2017.3 instead.	Reason: reason
6011-Deprecated	Remote login failure This event has been deprecated. Use Centrify Event Id 6034 introduced in release 2017.2 instead.	Reason: reason
6012-Deprecated	Run with privilege success This event has been deprecated. Use Centrify Event Id 6029 introduced in release 2017.2 instead.	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID Command: command PasswordPrompted: whether user was required to re-enter their password DesktopName: desktop name NetworkRoles: network roles

6013-Deprecated	Run with privilege failure This event has been deprecated. Use Centrify Event Id 6030 introduced in release 2017.2 instead.	Role: local role DesktopGuid: desktop GUID Command: command Reason: reason DesktopName: desktop name NetworkRoles: network roles
6014-Deprecated	Create desktop success This event has been deprecated. Use Centrify Event Id 6035 introduced in release 2017.2 instead.	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID PasswordPrompted: whether user was required to re-enter their password DesktopName: desktop name NetworkRoles: network roles
6018-Deprecated	Run with privilege failure This event has been deprecated. Use Centrify Event Id 6030 introduced in release 2017.2 instead.	Role: local role DesktopGuid: desktop GUID Command: command Reason: reason DesktopName: desktop name NetworkRoles: network roles PasswordPrompted: whether user was required to re-enter their password
6023	Leave from zone success	zone: zone name ZoneDomainName: zone domain name ComputerName: computer name ComputerDomainName: computer domain name LogonUser: logon user LogonUserSid: logon user SID AlternateUser: whether alternate user is used to perform the operation
6027	Add role assignment success	zone: zone name ZoneDomainName: zone domain name RoleName: role name Assignee: assignee LogonUser: logon user LogonUserSid: logon user SID AlternateUser: whether alternate user is used to perform the operation
6028	Add role assignment failure	zone: zone name ZoneDomainName: zone domain name RoleName: role name Assignee: assignee Reason: reason LogonUser: logon user LogonUserSid: logon user SID AlternateUser: whether alternate user is used to perform the operation
6029	Run with privilege success	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID Command: command PasswordPrompted: whether user was required to re-enter their password DesktopName: desktop name NetworkRoles: network roles EntityName: Entity Name MFARequired: whether user was required to do MFA
6030	Run with privilege failure	Role: local role DesktopGuid: desktop GUID Command: command Reason: reason DesktopName: desktop name NetworkRoles: network roles PasswordPrompted: whether user was required to re-enter their password EntityName: Entity Name MFARequired: whether user was required to do MFA
6031	Console login success	Role: role DesktopGuid: desktop GUID EntityName: Entity Name MFARequired: whether user was required to do MFA
6032	Console logon failure	Reason: reason EntityName: Entity Name MFARequired: whether user was required to do MFA
6033	Remote login success	Role: role DesktopGuid: desktop GUID EntityName: Entity Name MFARequired: whether user was required to do MFA
6034	Remote login failure	Reason: reason EntityName: Entity Name MFARequired: whether user was required to do MFA
6035	Create desktop success	Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID DesktopGuid: desktop GUID PasswordPrompted: whether user was required to re-enter their password DesktopName: desktop name NetworkRoles: network roles EntityName: Entity Name MFARequired: whether user was required to do MFA
6036	Create desktop failure	Role: local role Reason: reason NetworkRoles: network roles PasswordPrompted: whether user was required to re-enter their password EntityName: Entity Name MFARequired: whether user was required to do MFA

6037	Switch desktop success	DesktopName: desktop name DesktopGuid: desktop GUID PasswordPrompted: whether user was required to re-enter their password Role: local role NetworkRoles: network roles EntityName: Entity Name MFARequired: whether user was required to do MFA
6038	Switch desktop failure	DesktopName: desktop name Reason: reason PasswordPrompted: whether user was required to re-enter their password EntityName: Entity Name MFARequired: whether user was required to do MFA
6039	Network access success	Role: role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon GUID EntityName: Entity Name MFARequired: whether user was required to do MFA
6040	Self-service password reset success added in release 2017.3	Username: username
6041	Self-service password reset failure added in release 2017.3	Username: username Reason: failure reason
6042	Self-service account unlock success added in release 2017.3	Username: username
6043	Self-service account unlock failure added in release 2017.3	Username: username Reason: failure reason
6044	Enable Centrify Identity Services Platform succeeded added in release 2017.3	PlatformInstance: Platform Instance
6045	Disable Centrify Identity Services Platform succeeded added in release 2017.3	PlatformInstance: Platform Instance
6046	Enable Centrify Identity Services Platform failed added in release 2017.3	PlatformInstance: Platform Instance Reason: Reason for failure
6047	Disable Centrify Identity Services Platform failed added in release 2017.3	PlatformInstance: Platform Instance Reason: Reason for failure
6048	PowerShell remote connection success added in release 18.8	User: user Role: role
6049	PowerShell remote connection failure added in release 18.8	User: user Reason: reason
6050	Trouble ticket entered added in release 18.11	ticket: ticket reason: reason for privilege elevation comment: additional comment
		Role: local role EffectiveSid: effective user SID EffectiveGroupSids: effective group SID's LogonGuid: logon

6051	Run with privilege as an alternate user success added in release 18.11	GUID DesktopGuid: desktop GUID Command: command PasswordPrompted: whether user was required to re-enter their password DesktopName: desktop name NetworkRoles: network roles EntityName: Entity Name MfaRequired: whether user was required to do MFA AlternateUsername: An alternate username AlternateUserSid: An alternate user's SID
6052	Run with privilege as an alternate user failure added in release 18.11	Role: local role DesktopGuid: desktop GUID Command: command Reason: reason DesktopName: desktop name NetworkRoles: network roles PasswordPrompted: whether user was required to re-enter their password EntityName: Entity Name MfaRequired: whether user was required to do MFA AlternateUsername: An alternate username AlternateUserSid: An alternate user's SID
6053	Windows authentication is skipped added in release 18.11	service: service reason: Reason message for skip
6054	Run with alternate account success added in Release 2020	Command: command AlternateUsername: alternate username tenant: tenant URL PasswordPrompted: whether user was required to re-enter their password
6055	Run with alternate account failure added in Release 2020	Command: command AlternateUsername: alternate username tenant: tenant URL Reason: reason PasswordPrompted: whether user was required to re-enter their password
6300	Add roles and features success added in release 2018	PID: process id user: username@domain status: succeeded feature: feature name computer: computer name
6301	Add roles and features failure added in release 2018	PID: process id user: username@domain status: failed feature: feature name computer: computer name reason: reason for failure
6302	Remove roles and features success added in release 2018	PID: process id user: username@domain status: succeeded feature: feature name computer: computer name
6303	Remove roles and features failure added in release 2018	PID: process id user: username@domain status: failed feature: feature name computer: computer name reason: reason for failure
6350	Uninstall program success added in release 2018	PID: process id user: username@domain status: program: program name computer: computer name
6351	Uninstall program failure added in release 2018	PID: process id user: username@domain status: failed program: program name computer: computer name reason: reason for failure
6352	Change program success added in release 2018	PID: process id user: username@domain status: program: program name computer: computer name
6353	Change program failure added in release 2018	PID: process id user: username@domain status: failed program: program name computer: computer name reason: reason for failure
6354	Repair program success added in release 2018	PID: process id user: username@domain status: succeeded program: program name computer: computer name
6355	Repair program failure added in release 2018	PID: process id user: username@domain status: program: program name computer: computer name reason: reason for failure
6400	Enable network adapter success added in release	PID: process id user: username@domain status: succeeded adapter: adapter name computer: computer name

	2018	
6401	Enable network adapter failure added in release 2018	PID: process id user: username@domain status: failed adapter: adapter name computer: computer name reason: reason for failure
6402	Disable network adapter success added in release 2018	PID: process id user: username@domain status: succeeded adapter: adapter name computer: computer name
6403	Disable network adapter failure added in release 2018	PID: process id user: username@domain status: failed adapter: adapter name computer: computer name reason: reason for failure
6404	Rename network adapter success added in release 2018	PID: process id user: username@domain status: succeeded adapter: adapter name computer: computer name
6405	Rename network adapter failure added in release 2018	PID: process id user: username@status: failed adapter: adapter name computer: computer name reason: reason for failure
6406	Update IPv4 settings success added in release 2018	PID: process id user: username@domain status: succeeded adapter: adapter name computer: computer name
6407	Update IPv4 settings failure added in release 2018	PID: process id user: username@domain status: failed adapter: adapter name computer: computer name reason: reason for failure
6408	Update IPv6 settings success added in release 2018	PID: process id user: username@domain status: succeeded adapter: adapter name computer: computer name
6409	Update IPv6 settings failure added in release 2018	PID: process id user: username@domain status: failed adapter: adapter name computer: computer name reason: reason for failure
6500	Auto-enroll as corporate owned device success added in release 2018	computer: computer name tenant: tenant URL
6501	Auto-enroll as corporate owned device failure added in release 2018	computer: computer name tenant: tenant URL reason: reason for failure
6502	Unenroll device success added in release 2018	user: user name computer: computer name
6503	Unenroll device failure added in release 2018	user: user name computer: computer name reason: reason for failure
6504	Enroll as corporate owned device success added in release 2018	user: user name computer: computer name tenant: tenant URL
6505	Enroll as corporate owned device failure added in release 2018	user: user name computer: computer name tenant: tenant URL reason: reason for failure

6506	Enroll device success added in release 2018	user: user name computer: computer name tenant: tenant URL
6507	Enroll device failure added in release 2018	user: user name computer: computer name tenant: tenant URL reason: reason for failure
6508	Auto-unenroll success added in release 18.8	computer: computer name
6509	Auto-unenroll failure added in release 18.8	computer: computer name reason: reason for failure
6510	PowerShell remote command execution added in release 2020.1	userSid: User SID userName: User name authMechanism: Authentication mechanism url: HTTP URL of inbound request command: PowerShell remote command isScript: Command is a remote script

Centrify Authentication Service UNIX Agent

The Centrify Authentication Service UNIX Agent audit events are focused on the success or failure of starting and stopping the Centrify Agent: **adclient**.

Centrify Authentication Service UNIX Agent Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 17000. This log sample documents the successful start of the Centrify Agent: adclient. The change was made by user=root on April 05 at 06:46:43.

```
Apr 5 06:46:43 newcentos adclient[1837]: INFO AUDIT_
TRAIL|Centrify Suite|DirectControl UNIX Agent|1.0|2000
Centrify Agent (adclient) started|5|user=root pid=1837
utc=1459856803582 CentrifyEventID=17000
DAInst=AuditingInstallation DASessID=c72252aa-
e616-44ff-a5f6-d3f53f09bb67 status=SUCCESS service=adclient
```

Centrify Authentication Service UNIX Agent Audit Events

Authentication Service UNIX Agent Audit Events

17000	Centrify Agent (adclient) started	
17001	Centrify Agent (adclient) failed to start	reason: error message
17002	Centrify Agent (adclient) stopped	
17003	Centrify Agent (adclient) failed to stop	reason: error message

dzdo

For Linux and UNIX computers, Server Suite includes authorization services that enable users to run with elevated privileges using the dzdo command line program. The dzdo program is similar to sudo except that, instead of using a sudoers configuration file, the program uses the role-based access rights for zones stored in Active Directory.

dzdo Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 30004. This log sample documents that the dzdo service has been granted authorization.The change was made by user=dwirth(type:ad,dwirth@acme.vms) on April 7 at 01:20:12.

```
Apr 7 01:20:12 engcen6 adclient[2191]: INFO AUDIT_
TRAILICentrify Suiteldzdol1.0l0ldzdo
grantedl5luser=dwirth(type:ad,dwirth@acme.vms)
pid=32224 utc=1460010012602 Centrify EventID=30004
DAInst=AuditingInstallation DASessID=c72252aa-e616
-44ff-a5f6-d3f53f09bb67 status=GRANTED
service=dzdo command=/bin/vi runas=root role=ROLE_SYSTEM_
Archt/Global env=(none)
```

dzdo Audit Events

dzdo Audit Events

30000-Deprecated	dzdo granted This event has been deprecated. Use Centrify Event Id 30004 introduced in release 2017.3 instead.	command: command runas: username@domain role: role name env: environment variables
30001-Deprecated	dzdo denied This event has been deprecated. Use Centrify Event Id 30005 introduced in release 2017.3 instead. If the command is valid and requires authentication, Centrify Event Id 30005 is generated in release 2017.3 (and later versions) to show whether MFA is required or not.	command: command runas: username@domain reason: error message
30002	Trouble ticket entered	ticket: ticket
30004	dzdo granted added in release 2017.3	command: command runas: username@domain role: role name env: environment variables MfaRequired: whether user was required to do MFA EntityName: Entity Name
30005	dzdo denied added in release 2017.3	command: command runas: username@domain reason: error message MfaRequired: whether user was required to do MFA EntityName: Entity Name
30100	dzdo command execution starts added in release 18.11	command: command runas: username@domain role: role name env: environment variables MfaRequired: whether user was required to do MFA EntityName: Entity Name
30101	dzdo command execution ends added in release 18.11	

dzinfo

The dzinfo command displays rights, roles, and role assignments events. The dzinfo audit events focus on the success and failure of the dzinfo command.

dzinfo Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 42001. This log sample documents that a user failed run dzinfo to view another user's settings; only the user=root can view other user's settings. The change was made by user=eugene.user(type:ad,eugene.user@CENTSPLUNK.COM) on April 28 at 10:35:47.

```
Apr 28 10:35:47 sspl1-n2 adclient[1835]: INFO AUDIT_
TRAIL|Centrify Suite|dzinfo|1.0|3001|Dzinfo failed|5|user
=eugene.user(type:ad,eugene.user@CENTSPLUNK.COM)
pid=59947 utc=1461864947244 CentrifyEventID=42001
DAInst=AuditingInstallation DASessID=c72252aa-e616-
44ff-a5f6-d3f53f09bb67 status=FAILURE service=dzinfo
parameters=-c aaron.admin reason=Only root may view
other user's settings
```

dzinfo Audit Events

dzinfo Audit Events

42000	dzinfo successful	parameters: parameters
42001	dzinfo failed	parameters: parameters reason: error message

dzsh

For Linux and UNIX computers, Server Suite includes authorization services that enable users to run with elevated privileges in a restricted shell environment using the dzsh program.

dzsh Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 33001. This log sample documents a user being denied dzsh command execution.The change was made by user=dwirth(type:ad,dwirth@acme.vms) on April 7 at 01:20:12.

```
Apr 28 10:26:41 sspl1-n2 adclient[1835]: INFO AUDIT_
TRAILICentrify Suiteldzshl1.0l1ldzsh command execution
deniedl5luser=root pid=59860 utc=1461864401103 CentrifyEventID=33001
DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=DENIED service=dzsh command=/usr/share/
Centrifydc/bin/dzinfo reason=sam checking returned false,
user is not allowed to use this command or runas
```

dzsh Audit Events

dzsh Audit Events

33000-Deprecated	dzsh command execution granted This event has been deprecated. Use Centrify Event Id 33002 instead, which was introduced in release 2017.3.	command: command runas: username@domain role: role name env: environment variables
33001-Deprecated	dzsh command execution denied This event has been deprecated. Use Centrify Event Id 33003 instead, which was introduced in release 2017.3.	command: command reason: error message
33002	dzsh command execution granted added in release 2017.3	command: command runas: username@domain role: role name env: environment variables MfaRequired: whether user was required to do MFA EntityName: Entity Name
33003	dzsh command execution denied added in release 2017.3	command: command reason: error message MfaRequired: whether user was required to do MFA EntityName: Entity Name
34000	dzsh role change granted	fromRole: fromRole toRole: toRole
34001	dzsh role change denied	

License Management

Auditing licenses are issued for each computer that will be connected to an auditing collector, and are managed by the Centrify Licensing Service. You can use the Licensing Service control panel as described in the *License Management Administrator's Guide* to add and remove licenses, monitor license usage, and configure license usage notification.

License Management Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 20101. This log sample documents a user being denied an adjoin command execution due to missing license information. The change was made by user=root on October 27 at 17:24:25.

```
Oct 27 17:24:25 Eason5 adjoin[9886]: INFO AUDIT_
TRAIL|Centrify Suite|Centrify Commands|1.0|2101|Join
failed|5|user=root pid=9886 utc=1477560265956
CentrifyEventID=20101 DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=FAILURE service=adjoin parameters=-z developer
-p * eason.test zone=developer domain=eason.test
computer=eason5 runas=Administrator reason=Valid
Centrify license information was not found.
```

License Management Audit Events

License Management Audit Events

60100	authentication service license key added	PID: process id user: username@domain RunAs: username@domain type: user type key: license key container: license container
60101	Add authentication service license key failed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key container: license container reason: Error message
60102	authentication service license key removed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key container: license container
60103	Remove authentication service license key failed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key container: license container reason: Error message
60104	authentication service license container added	PID: process id user: username@domain RunAs: username@domain type: user type container: license container
60105	Add authentication service license container failed	PID: process id user: username@domain RunAs: username@domain type: user type container: license container reason: Error message
60106	authentication service license container removed	PID: process id user: username@domain RunAs: username@domain type: user type container: license container
60107	Remove authentication service license container failed	PID: process id user: username@domain RunAs: username@domain type: user type container: license container reason: Error message
60200	Add audit and monitoring service license key failed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key installation: installation reason: Error
60202	audit and monitoring service license key removed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key installation: installation
60203	Remove audit and monitoring service license key failed	PID: process id user: username@domain RunAs: username@domain type: user type key: license key installation: installation reason: Error message

Kerberos

Audit events in the Kerberos category are focused on the success or failure of kerberos credential access. Audit events are recorded when programs access the KCM (Kerberos Cache Manager) credential cache.

Kerberos Audit Event Log Sample

```
Sep 29 11:27:22 AbelRedhat5 adclient[8002]: INFO
AUDIT_TRAILICentrify SuiteIKerberosI1.0I200IInitializing
KCM credential cache succeededI5Iuser=root pid=8584
utc=1538191642025 CentrifyEventID=63200 DASessID=N/A
DAInst=N/A status=SUCCESS service=kcm process=adclient
pid=8002 ccache=1001 principal=user1@ABEL.TEST
```

Kerberos Audit Events

Kerberos Audit Events

63100	Generating new KCM credential cache name succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name
63101	Generating new KCM credential cache name failed	process: process name pid: process id reason: error message
63200	Initializing KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name principal: principal name
63201	Initializing KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name principal: principal name reason: error message
63300	Destroying KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name
63301	Destroying KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63400	Updating KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name principal: user principal services: service principal
63401	Updating KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63500	Retrieving credential in the given KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: ccache name
63501	Retrieving credential in the given KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63600	Reading principal in the given KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name principal: principal name
63601	Reading principal in the given KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63700	Iterating credentials in the given KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name

63701	Iterating credentials in the given KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63800	Reading credentials in the given KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name
63801	Reading credentials in the given KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
63900	Removing credentials from KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name principal: user principal services: service principal
63901	Removing credentials from KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
64000	Iterating KCM credential caches succeeded added in release 18.11	process: process name pid: process id
64100	Reading KCM credential caches succeeded	process: process name pid: process id
64101	Reading KCM credential caches failed added in release 18.11	process: process name pid: process id reason: error message
64200	Changing the ownership for the given credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name uid: uid gid: gid
64201	Changing the ownership for the given credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message
64300	Reading status for the given KCM credential cache succeeded added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name
64301	Reading status for the given KCM credential cache failed added in release 18.11	process: process name pid: process id ccache: Kerberos credential cache name reason: error message

Local Account Management

Centrify administrators use the Local Account Management feature to create, manage, lock, and delete local UNIX and Linux user and group accounts. The Local Account Management audit events focus on local users, groups, and accounts.

Local Account Management Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 51300. This log sample documents the removal of a local user from a local password file. The change was made by user=root on November 25 at 16:51:20.

```
Nov 25 16:51:20 rhed57x64v3 adclient[4423]: INFO
AUDIT_TRAIL|Centrify Suite|Local Account
Management|1.0|300|Removing local user from local passwd
file|5|user=root pid=4423 utc=1448441900487 CentrifyEventID=51300
DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=SUCCESS removedUser=locud01
```

Local Account Management Audit Events

Event Source Category: Local Account Management

51100	Adding enabled local user to local passwd file	enabledUser: enabled local user
51200	Adding disabled local user to local passwd file	disabledUser: disabled local user
51300	Removing local user from local passwd file	removedUser: removed local user
51400	Local user is marked as disabled	localUser: local user
51500	Local user is marked as enabled	localUser: local user
51101	Local passwd file update failed	reason: error message
51600	Invoking notification cli succeeded	parameters: parameters
51601	Invoking notification cli failed	reason: error message
52000	Adding enabled local group to local group file	enabledGroup: enabled local group
52100	Removing local group from local group file	removedGroup: removed local group
52001	Local group file update failed	reason: error message
53000	Managing local accounts succeeded	parameters: parameters
53001	Managing local accounts failed	parameters: parameters reason: error message
53100	Added enabled local user added in Release 2020	localuser: user name
53101	Added disabled local user added in Release 2020	localuser: user name
53102	Failed to add local user added in Release 2020	localuser: user name reason: error message
53103	Removed local user added in Release 2020	localuser: user name
53104	Failed to remove local user added in Release 2020	localuser: name reason: error message

53105	Enabled local user added in Release 2020	localuser: user name
53106	Failed to enable local user added in Release 2020	localuser: user name reason: error message
53107	Disabled local user added in Release 2020	localuser: user name
53108	Failed to disable local user added in Release 2020	localuser: user name reason: error message
53109	Modified local user added in Release 2020	localuser: user name
53110	Failed to modify local user added in Release 2020	localuser: user name reason: error message
53111	Added local group added in Release 2020	localgroup: group name
53112	Failed to add local group added in Release 2020	localgroup: group name reason: error message
53113	Removed local group added in Release 2020	localgroup: group name
53114	Failed to remove local group added in Release 2020	localgroup: group name reason: error message
53115	Modified local group added in Release 2020	localgroup: group name
53116	Failed to modify local group added in Release 2020	localgroup: group name reason: error message
53117	Managed local users and groups added in Release 2020	
53118	Failed to manage local users and groups added in Release 2020	reason: Reason for failure
53119	Invoked notification command added in Release 2020	command: notification command
53120	Failed to invoke notification command added in Release 2020	reason: Reason for failure

Multi-Factor Authentication

Multi-factor authentication (MFA) strengthens security by requiring users to provide more than one form of identification to authenticate their identity when they attempt to access servers or applications. Multi-factor authentication challenges might require users to type a password, respond to an email message or phone call, enter a passcode, or answer a security question. Audit events in the MFA category focus on the success and failure of MFA challenges.

Multi-Factor Authentication Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 54100. This log sample documents the success of an MFA challenge. The change was made by user=laniu1(type:ad,laniu1@SINGLE01.CDC) on April 20 at 14:51:18.

```
Apr 20 14:51:18 sol112x64v3 adclient[5640]: [ID 702911
auth.info] INFO AUDIT_TRAILICentrify SuiteIMFAI1.0
I100IMFA challenge succeededI5Iuser=laniu1(type:ad,
laniu1@SINGLE01.CDC) pid=6160 utc=1461135078139
CentryifyEventID=54100 DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=SUCCEED service=sshd tty=ssh client=::1
challenge=EMAIL
```

Multi-Factor Audit Events

MFA Audit Events

54100-Deprecated	MFA challenge succeeded This event has been deprecated. Use Centrify Event Id 54102 introduced in release 2017.3 instead.	service: service tty: tty client: client challenge: challenge
54101-Deprecated	MFA challenge failed This event has been deprecated. Use Centrify Event Id 54103 introduced in release 2017.3 instead.	service: service tty: tty client: client challenge: challenge reason: error message
54102	MFA challenge succeeded added in release 2017.3	service: service tty: tty authmethod: Reserved. factorcount: Number of MFA challenges factors: MFA challenges used. mfaresult: MFA challenge status. sourcehost: Remote host username: Username entityname: local system name devicetype: host operating system type initiatortype: MFA event type entitytype: event type description rolename: DirectAuthorize role used command: command used
54103	MFA challenge failed added in release 2017.3	service: service tty: tty authmethod: Reserved. factorcount: Number of MFA challenges factors: MFA challenges used. mfaresult: MFA challenge status. sourcehost: Remote host username: Username entityname: local system name devicetype: host operating system type initiatortype: MFA event type entitytype: event type description rolename: DirectAuthorize role used command: command used reason: error message
54200	MFA challenge succeeded	service: service challenge: challenge
54201	MFA challenge failed	service: service challenge: challenge reason: error message
54202	MFA is offline	service: service reason: error message
54203	MFA is skipped	service: service reason: message
54204	MFA challenge succeeded added in release 2017.3 This event has been deprecated. Use Centrify Event ID 54206 instead, which was introduced in release 2018.	service: service authmethod: authmethod factorcount: factorcount factors: factors mfaresult: mfaresult sourcehost: sourcehost username: username entityname: entityname entitytype: entitytype devicetype: devicetype rolename: rolename command: command

54205	MFA challenge failed added in release 2017.3 This event has been deprecated. Use Centrify Event ID 54207 instead, which was introduced in release 2018.	service: service reason: error message authmethod: authmethod factorcount: factorcount factors: factors mfaresult: mfaresult sourcehost: sourcehost username: username entityname: entityname entitytype: entitytype devicetype: devicetype rolename: rolename command: command
54206	MFA challenge succeeded Added in release 2018	service: service authmethod: authmethod factorcount: factorcount factors: factors mfaresult: mfaresult sourcehost: sourcehost username: username entityname: entityname entitytype: entitytype initiatortype: initiatortype devicetype: devicetype rolename: rolename command: command
54207	MFA challenge failed Added in release 2018	service: service reason: error message authmethod: authmethod factorcount: factorcount factors: factors mfaresult: mfaresult sourcehost: sourcehost username: username entityname: entityname entitytype: entitytype initiatortype: initiatortype devicetype: devicetype rolename: rolename command: command
54208	Setup MFA offline profile succeeded added in release 18.11	Username: The name of user configurationType: The MFA offline confguration type deviceType: The MFA offline device type
54209	Setup MFA offline profile failed added in release 18.11	Reason: The reason why it is failed Username: The name of user configurationType: The MFA offline confguration type deviceType: The MFA offline device type
54210	MFA challenge succeeded added in release 19.6	service: service authentication: authentication challenge: challenge
54211	MFA challenge failed added in release 19.6	

PAM

A pluggable authentication module (PAM) is a mechanism to integrate multiple low-level authentication schemes into a high-level application programming interface (API). The PAM audit events include authorization, credentials, account management, password changes, open session, and multi-factor authentication.

PAM Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 24100. This log sample documents PAM authentication being granted. The change was made by user=dwirth(type:ad,dwirth@acme.vms) on April 4 at 21:04:14.

```
Apr 4 21:04:14 engcen6 adclient[1749]: INFO AUDIT_
TRAIL|Centrify Suite|PAM|1.0|100|PAM authentication
granted|5|user=dwirth(type:ad,dwirth@acme.vms) pid=7458
utc=1459784054942 CentrifyEventID=24100
DAInst=AuditingInstallation DASessID=c72252aa-e616
-44ff-a5f6-d3f53f09bb67 status=GRANTED
service=sshd tty=ssh client=dc.acme.vms
```

PAM Audit Events

PAM Audit Events

24100-Deprecated	PAM authentication granted This event has been deprecated. Use Centrify Event Id 24102 introduced in release 2017.3 instead.	service: service tty: tty client: client
24101-Deprecated	PAM authentication denied This event has been deprecated. Use Centrify Event Id 24103 introduced in release 2017.3 instead.	service: service tty: tty client: client reason: error message
24102	PAM authentication granted added in release 2017.3	service: service tty: tty client: client MfaRequired: whether user was required to do MFA EntityName: Entity Name
24103	PAM authentication denied added in release 2017.3	service: service tty: tty client: client reason: error message MfaRequired: whether user was required to do MFA EntityName: Entity Name
24200	PAM set credentials granted	service: service tty: tty client: client
24201	PAM set credentials denied	service: service tty: tty client: client reason: error message
24300	PAM account management granted	service: service tty: tty client: client
24301	PAM account management denied	service: service tty: tty client: client reason: error message
24400	PAM change password granted	service: service tty: tty client: client
24401	PAM change password denied	service: service tty: tty client: client reason: error message
24500	PAM open session granted	service: service tty: tty client: client
24501	PAM open session denied	service: service tty: tty client: client reason: error message
24600	PAM close session granted	service: service tty: tty client: client
24601	PAM close session denied	service: service tty: tty client: client reason: error message
24700	The user logins to the system in rescue mode added in release 18.11	service: service tty: tty client: client

Delinea

Trusted Path

The trusted path configuration parameter (audittrail.Centrify_Suite.Trusted_Path.machinecred.skipda) specifies whether trusted path audit trail events are sent to the audit installation database in situations where the user is using a computer credential. The audit events identify a granted and denied Trusted Path.

Trusted Path Audit Event Log Sample

The following is a sample of an audit event log for Centrify Audit Event ID 23700. This log sample documents a Trusted Path being granted. The change was made by user=newcentos$@acme.vms on April 04 at 21:02:09.

```
Apr 4 21:02:09 newcentos adclient[1395]: INFO AUDIT
_TRAILICentrify SuitelTrusted Pathl1.0l2700lTrusted path
grantedl5luser=newcentos$@acme.vms pid=1395
utc=1459783929161 CentrifyEventID=23700 DAInst=AuditingInstallation
DASessID=c72252aa-e616-44ff-a5f6-d3f53f09bb67
status=GRANTED server=ldap/dc.acme.vms@acme.vms
```

> **Note**: The Trusted path audit event log sample identifies a server field type instead of the usual service field type found in UNIX/Linux audit events.

Trusted Path Audit Events

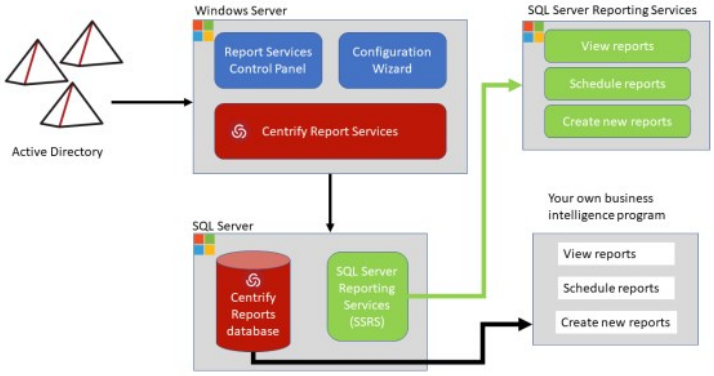
Trusted Path Audit Events

23700	Trusted path granted	server: server
23701	Trusted path denied	server: server reason: error message

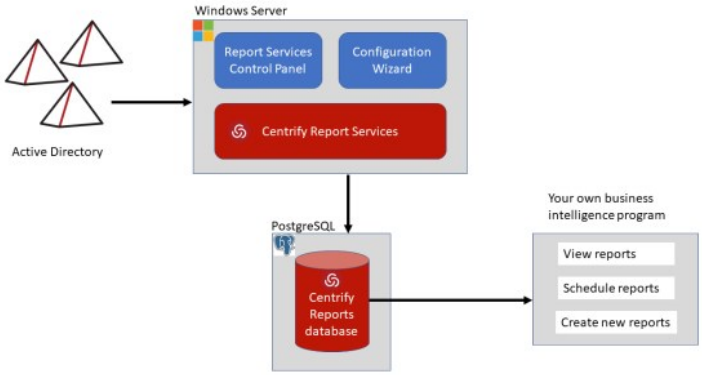
Report services provides reports on your Active Directory environment and the data is stored in a database that's optimized for reporting. You can synchronize your Active Directory information to your reporting database, and then allow your users access to the reporting data.

You can choose to use SQL Server or PostgreSQL for your report database. If you use PostgreSQL, you must provide your own report software to create and view reports.

If you're using SQL Server, the following diagram illustrates the main report services architecture components:



If you're using PostgreSQL, the following diagram illustrates the main report services architecture components:



Report services takes data from Active Directory at a particular point in time. The data collected at that point is sometimes referred to as a *snapshot*. The Active Directory data synchronization service puts the Active Directory data into tables in the reporting database, and then runs some algorithms on those tables. Some data is pulled over directly from Active Directory as it is, and some data is calculated.

For example, the effective role assignment for each computer and user is calculated rather than stored. Delinea does store the effective role assignment information at the levels of role, computer, and zone. This information is then stored in the database views, and those database views provide the information that you see in the reports.

The reporting service populates *database views* based on the data in those tables, and those views are what are used to populate reports.

Database views provide an easier and more secure way to share the reporting data without having to expose the database tables directly. Each view is essentially a database query. Some columns refer to columns in other views, and these relationships are noted.

Each default report is based on one or more of those database views, and you can build custom reports based on the information stored in one or more of those views.

For SQL Server databases, Delinea report services uses Microsoft SQL Server Reporting Services as the reporting engine for deploying and customizing

reports. You can use any reporting service to generate reports by connecting to the reporting database.

Reporting Data Based on Domains or Zones

Here are some key points to be aware of if you're thinking of using report data based on zones:

- For zone-based reporting, each synchronization includes all Active Directory data from the specified zones. In comparison, for domain-based reporting, synchronizations after the first one include just the changes to Active Directory data.
- For zone-based reporting, the service account needs just read permission to Active Directory. In comparison, for domain-based reporting, the service account needs permission to replicate directory changes.
- For zone-based reporting, report services does not synchronize license information nor auto-zone computer information.
- For zone-based reporting, you can include zones from other trusted forests. For domain-based reporting, you can add trusted forest domains.

gMSA Accounts

Report services treats gMSA accounts (group Managed Service Accounts) as Active Directory users.

Information not included in the reporting database

There are few limitations on the kinds of data that can be stored in the reporting database. The following is not included:

- NIS maps
- UNIX import information

Report Services and Report Center

Report services provides more reports and features than the previous Report Center in Server Suite. Report Center has been deprecated and removed.

Report Services Tools Overview

Here's an overview of the tools specific to Delinea report services. You'll use some to all of these tools, depending on whether you're completing your initial installation or changing some configuration settings later on.

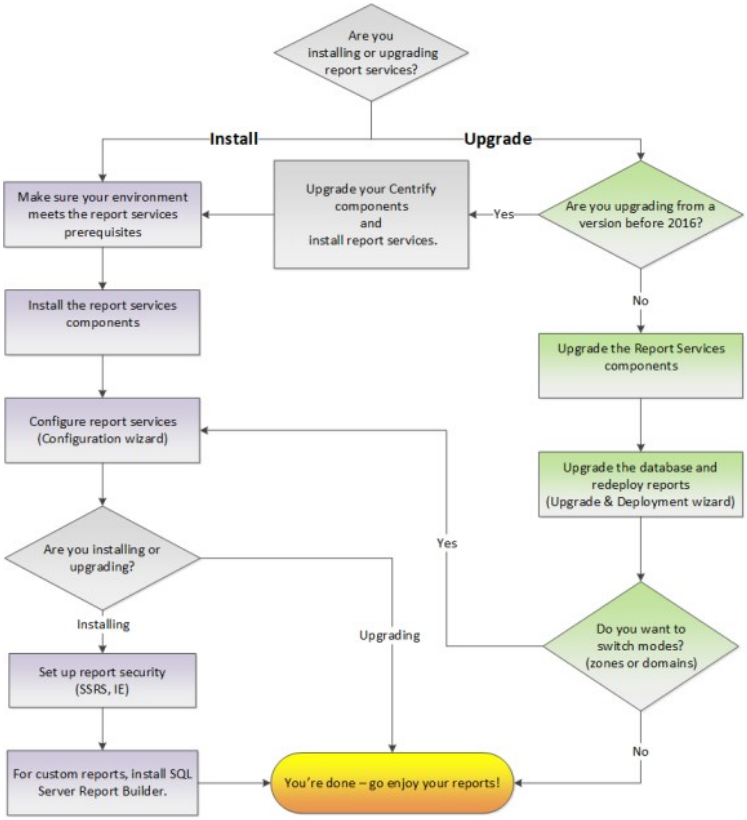
Report Services shortcut	Use this shortcut to open Delinea report services in Internet Explorer.
Configuration wizard	Use the configuration wizard to do the initial setup of your database and reports. Re-run the configuration wizard only if you need to change some report services configuration settings or change whether you gather report data from Active Directory based on zones or domains. For instructions, see Configuring Report Services and Deploying Your Reports .
Upgrade & Deployment wizard	Use the Upgrade & Deployment wizard to upgrade your report database and deploy updated reports. For instructions, see Upgrading your report services.
Report Services Control Panel	Use the control panel to view the synchronization status of domains or zones, refresh report data, configure the synchronization schedule, add or remove domains or zones, change the user account that runs the report service, and view error logs. For more details, see Administering Delinea Report Services with the Report Control Panel.
Server Suite installer	Use the installer to either install or upgrade the report services and other Server Suite components. For instructions, see Installing Delinea Report Rervices .

Overview of How to Set Up Reporting

If you're installing an evaluation version of Delinea report services, you can take a few shortcuts, such as using virtual machines. This section includes

recommendations for both evaluation and production deployments.

The diagram below outlines the overall process for installation or upgrade.



Evaluation Deployment Overview

For evaluation purposes, you can just install the SQL Server Express version that's packaged with the Server Suite software.

How to set up an evaluation version of Delinea report services:

1. Prepare your environment:
    - Users and groups with required permissions
        1. service account - the user account that runs the reporting service (in the background)
        2. installer/administrator - the user account that installs and configures the Delinea reporting service.
        3. Report administrator - user(s) who can run reports, edit reports, build new reports
        4. Report reader - user(s) who can view reports but not edit them nor create new ones.
    - An existing database instance, if you're planning to use an existing instance.
    - The correct operating system that supports what you need. For evaluation purposes only, you can install all the software on one computer. Be sure to check that your operating system is supported for Delinea software, SQL Server, and Microsoft SQL Server Reporting Services (SSRS).
    - You've configured Internet Explorer to allow access to the reporting web site. For details, see Adding Your Report Services Web Site to your Internet Explorer Trusted Sites.
2. Run the Delinea installer. Install the report services on ONE computer in your domain.
    - Do not install Delinea report services on a domain controller.
    - If you're upgrading from a prior version of Server Suite or Server Suite, the Access Manager reports are still there and they are installed anywhere you install Access Manager. In contrast, the new reporting service installs into one place in your forest. Plus, the database is optimized for reporting and retrieval.
3. Do the reporting configurations:
    - Run the Report Services Configuration wizard to configure the reporting service as needed, including starting the service.
    - Set up the report security for report administrators by assigning users and groups to SSRS roles. By default, all authenticated users have access to view reports.

    - Configure Internet Explorer.
4. View and share the reports.
5. For custom report building, make sure that you've installed Report Builder for your version of SQL Server, if you don't have it installed already. You may need to download this separately.

Production Deployment Overview

For production deployments:

- Delinea recommends that you use a production-capable version of SQL Server and not SQL Server Express.

  SQL Server Express has a limit of 10Gb of data, does not provide the ability to schedule tasks

- Delinea recommends that you do not use virtual machines.
- Use at least 4 GB memory and 2 cores. leave enough memory for the operating system and allocate the rest to SQL server. For more details, see Memory Requirements.
- Delinea recommends that you use a new database instance; do not use an existing instance of SQL server. The reason for this is because uninstalling SSRS leaves some files behind and can cause problems with re-installation, if you're reusing the database instance. For more information, see Impact of Using a New or Existing SQL Server instance.
- If you're using a PostgreSQL database, Delinea recommends using a new PostgreSQL installation.
- Do not install Delinea report services on a domain controller.

How to Set up a Production Version of Delinea Report Services

1. Prepare your environment:
    - Users and groups with required permissions. For details, see Before Installing - Prerequisites.
        1. service account - the user account that runs the reporting service (in the background)
        2. installer/administrator - the user account that installs and configures the Delinea reporting service.
        3. Report administrator - user(s) who can run reports, edit reports, build new reports
        4. Report reader - user(s) who can view reports but not edit them nor create new ones.
    - The correct operating system that supports what you need. The operating system needs to be supported for Delinea software, SQL Server, and SQL Server Reporting Services (SSRS).

      Don't install SSRS on the domain controller.

      IMPORTANT: Use an existing database instance with a real version of SQL Server, not the Express version. Express isn't designed to handle the performance needs of a production environment.
2. Run the Server Suite installer. Install the report services in ONE place in your forest.
    - If you're upgrading from a prior version of Server Suite, the Access Manager reports are still there and they are installed anywhere you install Access Manager. In contrast, the new reporting service installs into one place in your forest. Plus, the database is optimized for reporting and retrieval.
3. Do the reporting configurations:
    - Configure the reporting service as needed, including starting the service.
    - Set up the report security: assign users and groups to SSRS roles and configure Internet Explorer.
4. View and share the reports.
5. For custom report building, make sure that you've installed Report Builder for your version of SQL Server, if you don't have it installed already. You may need to download this separately.

Upgrade Overview

How to upgrade Delinea Report Services:

1. If you're upgrading from a version of Server Suite before version 2016, you need to install the report services components after you upgrade the other components.

   For details, see Upgrading from a Prior Version.

2. Run the installer program to upgrade your report services components.

   For details, see Upgrading from a Prior Version and the *Upgrade and Compatibility Guide*.

3. Upgrade the report database and, if you're ready to do so, redeploy your reports.

   For details, see Upgrading your Report Services Database.

4. (Optional) If you want to switch from domain-based reporting to zone-based reporting, or the other way around, run the Configuration wizard to switch modes.

   This step is optional and you can do switch modes at any time, not just during upgrade.

   For details, see Configuring Report Services and Deploying Your Reports.

Using this Guide

The guide provides the following information:

- What Report Services Provides provides an overview of the report services features and tools, including deployment overviews for production and evaluation deployments.
- Installing and Configuring Delinea Report Services provides detailed instructions for installing, upgrading, and configuring report services.
- Viewing Default Reports covers how to open a report, and provides some basic information on each of the default reports.
- Building Custom Reports provides some information about how to build your own, custom reports.
- Views to Use in Custom Reports lists the database views that you can use to populate your custom reports.
- Configuring Report Services for Large Active Directory Environments provides helpful information unique to large deployments.
- Troubleshooting Reports provides some helpful tips with common installation or configuration issues.
- Synchronized Active Directory attributes for Reports lists the object attributes that report services synchronizes from Active Directory.

Installing and Configuring Report Services

This section includes the following topics:

- Before Installing - Prerequisites
- Installing Delinea Report Services
- Configuring Report Services and Deploying Your Reports
- Doing a Silent Install and Configuration
- Upgrading from a Prior Version
- Administering Delinea Report Services with the Report Control Panel
- Configuring SQL Server Reporting Services (SSRS)
- Re-deploying SQL Server Reports To SSRS

  Note: If you are deploying into a large Active Directory environment, be sure to also read Memory Recommendations and Requirements for large Active Directory environments.

Before Install: Prerequisites

Note: For the full set of platform requirements, please visit this web page in the Delinea Technical Support area:

https://www.centrify.com/support/whats-new/infrastructure-services/

Supported Versions of SQL Server and SSRS

To use Delinea report services, you need to use a SQL Server that is one of the following versions:

- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2012 R2
- SQL Server 2014
- SQL Server 2016

For Microsoft SSRS, use the version that correlates with your SQL Server version. For example, if you're using SQL Server 2012 R2, then use Microsoft SSRS version 2012 R2.

Note: If you choose to use a version of SQL Server that requires .NET version 3.5 SP1, be sure to install .NET before configuring report services.

Note: If you run Report Services with Microsoft SQL Server 2012 Service Pack 2 and Visual Studio 2010 on the same system, please update Visual Studio 2010 to Service Pack 1. (Ref: CS-38553a)

Supported Versions of PostgreSQL

Delinea Report Services works with PostgreSQL databases that are version 11 or later.

Supported Browser Versions

Use the web browser versions that Microsoft supports for use with SQL Server Reporting Services, as mentioned in this page:

https://msdn.microsoft.com/en-us/library/ms156511.aspx

For Internet Explorer, the version of SQL Server and SQL Server Reporting Services (SSRS) that you use also determines which version of Internet Explorer is compatible with your deployment. Please consult the Delinea Knowledge Base article KB-6671 for details about which version of Internet Explorer you should use.

Required User Permissions for Report Services

Before you install Delinea report services, be sure you have the appropriate software and user accounts, which includes the following:

- Users with required permissions. Before installation, you must have users to run the Delinea installer.
- Report service account
- SQL Server service account (this is needed if you're installing using an existing instance)
- User accounts that can run the Report Configuration Wizard and the Reporting Control Panel. There are a few user accounts that you need to set up for use with Delinea report services. Here is a summary of the user accounts that you need to create and the permissions you need to explicitly grant.

Report Services Account Permissions

	For domain-based reporting: Replicating directory changes

report service account to run the Reporting Service	at the domain level (ADUC) and replicate directory changes in ADSI For zone-based reporting: Read permission	Log on as a service		
SQL Server service account to run SQL Server	n/a	Log on as a service		member of the securityadmin role
PostgreSQL service account				the account must have permission to connect to PostgreSQL and create a database
report admin to run the Report Configuration wizard or the Upgrade & Deployment wizard and deploy reports to an existing SQL Server instance	needs to be a member of the domain	n/a	Folder Settings > Content Manager role	member of the securityadmin role (At the very least, the user needs permission to connect to SQL Server and create a database.)
report admin to modify the Reports Control Panel	Read permission to the domain root object of the selected domain. Read permission to all computer objects in the selected domain.	n/a		
Report viewer to view reports from SSRS/Internet Explorer			Site settings > System user role Folder settings > browser (assign SSRS roles to Active Directory group or users)	
Report writer read, write, edit access for reports, in addition to the permissions needed to view reports			Site settings > System user role Folder settings > Content Manager role (assign SSRS roles to Active Directory group or users)	

> **Note**: Delinea Report Services requires administrator permission to install and upgrade. That also means that only an administrator can uninstall and repair Delinea Report Services. (Ref: CS-40808a)

Grant the Report Service Account Permissions

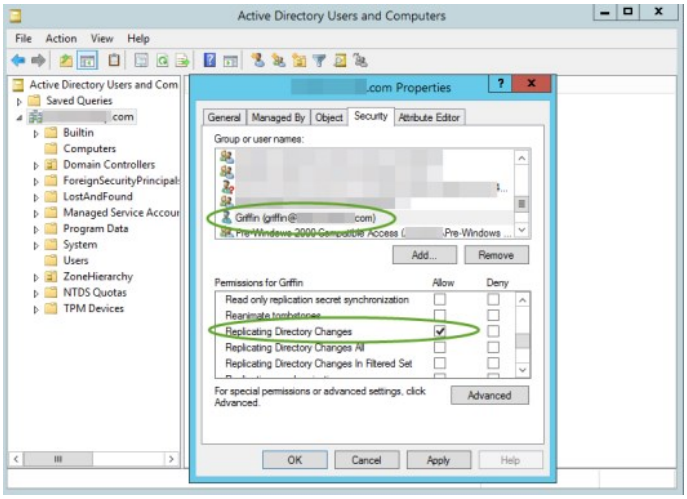
For your convenience, below are reminders for how to grant the two sets of required permissions for the report service account.

Grant the Permission to Replicate Directory Changes in ADUC

To grant the permission to replicate directory changes at the domain level (read only):

1. Open Active Directory Users and Computers.
2. From the View menu, select **Advanced Features**.

3. Right-click the domain object and select **Properties**.
4. Click the **Security** tab.
5. Select the desired user account (add the account if it's not listed there already).
6. In the Permissions area, next to **Replicating Directory Changes**, click **Allow**.



7. Click OK to save your changes.

   For more information about setting this permission, see https://support.microsoft.com/en-us/kb/303972.

Grant the Permission To Replicate Directory Changes In ADSI

In addition to granting the replicate directory changes permission in Active Directory Users and Computers (ADUC), you also need to grant the same permission in the ADSI Edit (Active Directory Services Interfaces Editor) console.

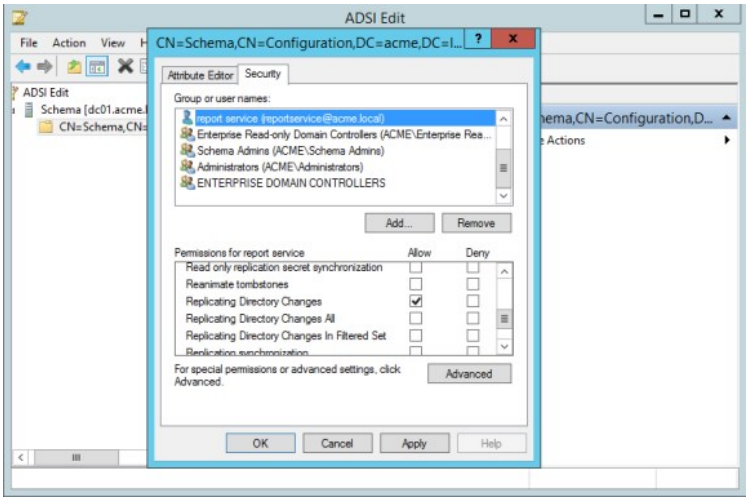
To grant the permission to replicate directory changes in ADSI (read only):

1. Open the ADSI Edit console.
2. From the Action menu, select **Connect to**.

   The Connection Settings dialog box opens.
3. For the Connection Point, go to the "Select a well known Naming Context" drop-down menu and select **Schema**.
4. Click **OK** to close the dialog box.

   The schema for the current domain displays in the ADSI Edit console.
5. Expand the schema listing so that you can see the first node of the schema, and right-click that node and select **Properties**.

   The Attribute Editor dialog box opens.
6. Click the **Security** tab.
7. Select the desired user account (add the account if it's not listed there already).
8. In the Permissions area, next to **Replicating Directory Changes**, click **Allow**.



9. Click OK to save your changes.

Grant the Permission to Log on as a Service

To grant the log on as a service permission:

1. In the Group Policy Management Editor, apply the following policy to your desired user or group of users:

   Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment > Log on as a Service.

For more details about granting the log on as a service policy, see https://technet.microsoft.com/en-us/library/dn221981(v=ws.11).aspx.

SQL Server permissions that are set by the Configuration Wizard

Here are the SQL server permissions that report services grants to each user type, for your information. The Report Services Configuration wizard sets these permissions automatically.

Sql Server Permissions Set by the Report Services Configuration Wizard (table)

report services account to run the SQL Server Reporting Service	Snapshot Service (predefined role)
SQL Server service account to run SQL Server	If you deploy to an existing SQL Server instance, the configuration wizard makes no changes to the SQL Server service account. If you deploy to a new SQL Server instance: --If the operating system is Windows 2008 and you're using a SQL Server version later than 2012, virtual accounts are used for various SQL Server components, as follows: SQL Server engine: NT SERVICE\MSSQL$<InstanceName> SQL Server Agent: NT SERVICE\SQLAgent$<InstanceName> Full text search: NT SERVICE\MSSQLFDLauncher$<InstanceName> SSRS: NT SERVICE\ReportServer$<InstanceName> --Otherwise, the SQL Server service accounts are configured as follows: SQL Server engine: NT Authority\Network Service SQL Server Agent: NT Authority\Network Service Full text search: NT Authority\Local Service SSRS: NT Authority\Local Service
report admin to run the Report	

Configuration Wizard and deploy reports to an existing SQL Server instance	Connect SQL (cannot be revoked after setup) Create Database, Create any database, or Alter any database member of securityadmin role, or Alter any login permission
report admin to modify the Reports Control Panel	SnapshotAdmin (predefined role)
Report viewer to view reports from SSRS/Internet Explorer	Login permission SnapshotViewer (predefined role)
Report writer read, write, edit access for reports, in addition to the permissions needed to view reports	Login permission SnapshotViewer (predefined role)

> **Note**: Microsoft SQL Server Reporting System (SSRS) affords only role-based security in their reports. Be sure to grant appropriate access to reports. For example, if a user has access to only some data in the specified domain but all reports, they will be able to view all reports on all data from Active Directory.

PostgreSQL Permissions that are Set by the Configuration Wizard

When you create the PostgreSQL database with the Configuration wizard, the wizard grants the administrator user one permission for Create Database.

Memory Requirements

Be sure that your computers running the following components meet or exceed the RAM requirements listed below.

Domain Controller Memory Requirements

The minimum amount of RAM that you should have available for your domain controller is the sum of the following:

- Active Directory database size (for example, C:\Windows\NTDS\)
- Total SYSVOL size (for example, C:\Windows\SYSVOL)
- Recommended amount of RAM for your operating system
- Vendor recommended amount of RAM for your various background software agents, such as anti-virus, monitoring, backup, and so forth.
- Additional RAM to accommodate growth over the lifetime of the server.

For more information, see Microsoft recommendations here: http://social.technet.microsoft.com/wiki/contents/articles/14355.capacity-planning-for-active-directory-domain-services.aspx.

Windows Memory Requirements

Here are the minimum and recommended memory requirements for report services and the report database:

- Delinea report services: minimum 2 GB RAM, recommended 4 GB or above

- SQL Server report database: minimum 4 GB RAM, recommended 32 GB or above
- PostgreSQL report database: minimum 4 GB RAM, recommended 32 GB or above

SQL Server Recovery Model Requirement

In order for report services to function efficiently, it's recommended that you configure your SQL Server database to use the Simple recovery model. The recovery model configuration determines how SQL Server logs transactions, whether a database backs up the transaction log, and what kinds of restore options are available.

For more information about recovery models, please visit https://msdn.microsoft.com/en-us/library/ms189275.aspx.

To configure the SQL Server database recovery model:

1. In SQL Server Management Studio, navigate to the database that you use for report services.
2. Right-click the database and select **Properties**.
3. In the Select a Page area, click **Options**.
4. For the Recovery Model option, select **Simple**.
5. Click **OK** to save the changes.

Impact of Using a New or Existing SQL Server Instance

When you set up your installation of Delinea report services, you have the option of either using an existing SQL Server instance or installing a new instance. Delinea recommends that you use a new SQL Server instance, if possible.

If you choose to install a new instance from the Delinea Management Services installer program, the program installs an instance of SQL Server Express 2008 R2 with Advanced Services.

If you have an existing installation of SQL Server, you can create a new instance there first on your own, using your own installation media. When you install or configure Delinea report services, you then configure report services to use your existing instance that you created. That way your SQL Server instances use the same edition and version.

Tip: {/b}Please see the information at the following link for details about installing multiple versions and instances of SQL Server:

https://msdn.microsoft.com/en-us/library/ms143694(v=sql.130).aspx

Here are some issues to be aware of if you're going to use a *new* SQL Server instance:

- With a new SQL Server instance, you can avoid any potential problematic issues with SSRS, particularly if you need to reinstall SSRS.
- SSRS won't slow down the regular database operations on other instances.
- To prevent the SQL Server instance from consuming too much memory, it's recommended to use the max server memory to control each SQL Server instance's memory usage. The total allowance is not more than the total physical memory on the machine. If user is not running all of the instances, none of the running instances will be able to utilize the remaining free memory.

Here are some issues to be aware of if you're going to use an *existing* SQL Server instance:

- There can be issues with SSRS and existing instances. If you have to uninstall and reinstall SSRS, it leaves files behind with the existing instance.
- Using an existing SQL server instance can use all the free memory with a larger limit of the max server memory setting.

If you choose to deploy report services using an existing instance of SQL Server, your database administrator may need to know what changes that report services needs to make to the database. (KB-8042)

The only modification that report services makes to an existing database instance is to add two Windows integrated logins, as follows:

<The specified service account>	SnapshotService
[NT Authority\Authenticated Users]	SnapshotViewer

> **Note**: If these logins already exist, report services does not re-create them.

Deploy in Multi-Forest Environments

If you're deploying report services across multiple forests, there are a few tips to be aware of.

- It is best to install report services once in a forest, and then monitor domains or zones in other trusted forests.
- If you use domain-based mode, you need to install report services once in the domain. Make sure that any users who run report services and the service account have access to the domains for which you want to run reports.

  > **Note**: If you need to grant access to a user account across a forest with a one-way selective trust, you enable selective authentication for that user.

Enable Selective Authentication Across a Forest with a One-Way Selective Trust

The instructions below are provided as a courtesy; for more information on selective authentication, see the following article:

https://technet.microsoft.com/en-us/library/cc794747(v=ws.10).aspx

How to enable selective authentication for a user across an Active Directory forest that has a one-way selective trust:

1. Open **Active Directory Domains and Trusts.**
2. In the console tree, right-click the domain node for the forest root domain, and then click **Properties**.
3. On the **Trusts** tab, under either **Domains trusted by this domain** (outgoing trusts) or **Domains that trust this domain** (incoming trusts), click the forest trust that you want to administer, and then click **Properties**.
4. On the **Authentication** tab, click **Selective authentication**, and then click **OK**.
5. Open **Active Directory Users and Computers**.
6. Navigate to the Domain Controller the Report Services will use, right-click the computer object, and then click **Properties**.
7. On the Security tab add the desired user and grant Allow for the **Allowed to authenticate** permission.

See also the Delinea knowledge base article KB-8071.

Virtual Machines and Report Services

For production deployments, it is recommended to avoid using virtual machines for use with report services. (KB-7038)

In general, report services works well in virtual machines, including the case of installing SQL Server in a virtual machine.

However, in a large enterprise environment (such as where more than 100,000 users are enabled for authentication service), the SQL queries used for generating reports may have significant CPU, memory and I/O requirements. In these situations, Delinea recommends the use of physical machines for SQL Server to allow for better tuning of SQL Server without impacting other systems.

Alternatively, you can install SQL Server in a virtual machine. In such cases, Delinea recommends that you follow the guidelines provided by the virtualization vendors:

https://www.vmware.com/files/pdf/solutions/SQL_Server_on_VMware-Best_Practices_Guide.pdf

http://download.microsoft.com/download/6/1/d/61dde9b6-ab46-48ca-8380-d7714c9cb1ab/best_practices_for_virtualizing_and_managing_sql_server_2012.pdf

Installing Report Services

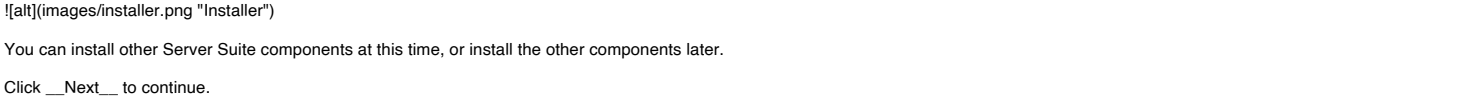
You use the same installer to install report services that you use to install other Server Suite components.

To install Delinea Report services:

1. Run the Delinea Management Services installer program that's appropriate for your Windows system (64-bit only).
2. In the Getting Started screen, click **Access**.
3. In the Welcome screen, click **Next** to continue.
4. Review the license agreement, and click the option that indicates that you agree to the terms.

   Click **Next** to continue.
5. In the User Registration screen, enter your name and company name.

   Click **Next** to continue.
6. Select the **Centrify Report Services** item.



7. In the Choose Destination Folder screen, specify the folder you want to install the software.

   If you're also installing Access Manager, you can select the options to automatically install desktop shortcuts.

   Click **Next** to continue.

   In the Confirm Installation Settings screen, review the list of components that will be installed. If the list is correct, click **Next** to continue.

   The program installs the files.
8. In the completion screen, select **Configure Report Services** and click **Finish**. Proceed to the next section, Configuring report services and deploying your reports .
9. If you don't want to configure report services right now, deselect the **Configure Report Services** option and click **Finish**. You can run the configuration wizard later, if desired.

Configuring Report Services and Deploying Your Reports

You use the configuration wizard to both set up a new report services deployment or reconfigure an existing one.

If you want to just redeploy your reports, see Re-deploying SQL Server reports to SSRS.

Configuring a SQL Server Report Services Deployment

Follow these instructions if you're creating a new report services deployment using SQL Server or reconfiguring an existing SQL Server report services deployment.

To configure report services with a SQL Server database:

1. If you need to start the Delinea Report Services configuration wizard, go to the **Start** menu > **All Programs** > **Centrify Server Suite 2021.1** > **Report Services**, and choose **Configuration Wizard.**

   If you're continuing from the Delinea Management Services installer, the installer started the configuration wizard for you.

2. On the Welcome screen, click **Next** to continue.
3. If you have already set up report services, the Reconfiguring Report Services screen displays. Select **Reconfigure** and click **Next** to continue.
4. On the Database Type screen, select **SQL Server** and click **Next** to continue.
5. Configure the SQL Server database connection:
   1. Specify the SQL Server instance name.

      Either specify a new SQL Server instance name, or select an existing SQL Server instance name. (The default instance name is CENTRIFYSUITE.)

      The SQL Server instance name must be 16 characters or less, the name cannot begin with an underscore (_) or dollar sign ($), and the instance name cannot contain any of the following special characters: a blank space, backslash (\), comma (,), colon (:), semi-colon (;), single quotation mark ('), ampersand (&), hyphen (-), number sign (#), or at sign (@).

      If you select an existing SQL Server instance, be aware that the SQL Server browser service must be running if SQL Server is a named instance or using dynamic ports. If for some reason the SQL Server service can't be started, you need to provide the SQL Server instance name and port number in order to connect to the database successfully. For additional details, see https://technet.microsoft.com/en-us/library/ms181087(v=sql.105).aspx.

      Delinea recommends that you use a new SQL Server instance, if possible. For more information, see Impact of using a new or existing SQL Server instance.

   2. The default database name is *Report*. You can change this, if desired.

      The SQL Server database name must be 16 characters or less, the name cannot contain any of the following special characters: backslash (\), forward slash (/), colon (:), asterisk (*), question mark (?), double quotes ("), less-than sign (<), greater-than sign (>), pipe (|), comma (,) or single quotation mark (').

   3. Click **Next** to continue.
   4. If you selected to install a new SQL Server instance, click **Browse** to navigate to and specify the location of the SQL server installation executable (*.exe file).

      The installer program installs SQL Server 2008 R2 Express with Advanced Services.

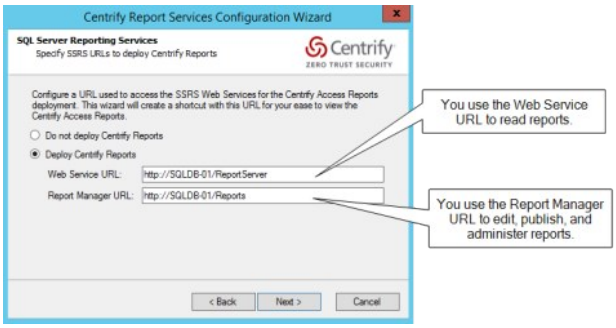
      You can download the SQL Server Express with Advanced Services package directly from Delinea, for your convenience. Or, download the package from Microsoft.

      Please ensure to download the file name SQLEXPRADV_x64_ENU.exe (1,008.6 MB in size) as this is the one containing the 64-bit edition of SQL 2008 R2 with the necessary additional components to support Delinea Reporting Services.

   5. Click **Next** to continue.
6. Deploy the reports:

1. In the SQL Server Reporting Services screen, specify whether to deploy the Server Suite reports (or not).

   If you plan to use a reporting solution other than Microsoft SQL Server Reporting Services, do not deploy the reports.

   This screen also lists the URLs for the Reporting Web Service and Report Manager. You'll use these URLs later to access to the reports.

   If you're using a production server of SQL Server and SSRS, you can configure them to use HTTPS. For details, see Microsoft SQL Server and SSRS documentation, such as https://msdn.microsoft.com/en-us/library/ms345223.aspx.

   The configuration wizard populates the report URLs automatically. If you had specified to use an existing SQL Server instance, the configuration wizard retrieves the existing web service URL and report manager URL for your SQL Server instance.

   For an existing SQL Server instance, you can open the Microsoft Reporting Services Configuration Manager to view the Web Service and Report Manager URLs.

2. Click **Next** to continue.

7. Choose domain or zone reporting:

   Specify whether you want to choose data for reporting based on domains or zones. The default is domain-based reporting.

   Click **Next** to continue. If you selected domain-based reporting, proceed to the next step. For zone-based reporting, go to Step 8.

8. If you selected domain-based reporting:

   1. In the Monitored Domain(s) screen, you can review and edit the list of domains that will be included for reporting. Add or remove domains as desired.

      For each domain, the configuration wizard lists the domain name and the domain controller name.

   2. Click **Next** to continue.

9. If you selected zone-based reporting and you use hierarchical zones:

   1. If you want data from all zones, select **Monitor all hierarchical zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
   2. Or, if you want to report data from specific zones, select **Monitor only specific hierarchical zones**.
   3. Click **Edit**.

      The Specify Forest for zone selection dialog box opens.

   4. Enter the forest name where the desired zones are and click **OK**.

      The Edit Monitored Hierarchical Zones dialog box opens.

   5. Enter the hierarchical zones by name, or expand the list of zones to locate the desired zones manually.
   6. If desired, specify to select the parent or child zones automatically.
   7. Select the zone by putting a checkmark in the box next to the zone name.

8. When you're done specifying which hierarchical zones to monitor, click **OK** to close the dialog box and return to the Configuration wizard.

   Each zone that you've selected is listed in the Hierarchical Zones screen.

9. Click **Next** to continue.

10. If you selected zone-based reporting and you use classic zones:

    1. If you want data from all zones, select **Monitor all classic zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
    2. Or, if you want to report data from specific zones, select **Monitor only specific classic zones**.
    3. Click **Edit**.

       The Specify Forest for zone selection dialog box opens.

    4. Enter the forest name where the desired zones are and click **OK**.

       The Edit Monitored Classic Zones dialog box opens.

    5. Select the classic zones to include in your reports. Select the zone by putting a checkmark in the box next to the zone name.

       You can filter the list of zones by entering a portion of the name and clicking **Filter**.

    6. When you're done specifying which classic zones to monitor, click **OK** to close the dialog box and return to the wizard. Click **Next** to continue.

11. For zone-based reporting, you can also specify which domain controller(s) that the report service connects to.

    If you don't specify which domain controller(s) to use, report services will use the default domain controller.

    1. Click **Add**.

       The Add Domain Controller dialog box opens.

    2. Enter the domain name and then select the domain controller from the list.
    3. Click **OK** to return to the Configuration wizard.

       The domain controllers that you selected are listed in the wizard screen.

    4. Click **Next** to continue.

12. In the Synchronization schedule screen, specify how often you want the reporting service to pull data from Active Directory.

    You can specify that the service synchronizes weekly, daily, every certain number of days, or every certain number of hours. The limit is 32,767 days or weeks.

    Click **Next** to continue.

13. Configure the user account that runs the service:

    1. In the Report Services options screen, specify the user account that will be used to run the service that synchronizes data from Active Directory and the reporting database.

       You can select a network service account, a managed service account, or another user account in Active Directory.

       You must specify a user account that has the required permissions. The configuration wizard verifies that the user has the correct level of access.

    2. Click **Next** to continue.
    3. The configuration wizard verifies that the specified user account has the required permission. An error displays if the permissions are inadequate.
    4. If the permission verification is successful, click **Close** to close the Verify permission window.

14. Review and complete the installation:

1. In the Summary screen, review the installation details. If the installation settings are correct, click **Next** to continue.

   If you're installing a new database, it may take a few minutes.

2. (Optional) In the completion screen, if the installation is successful, you can select the option to synchronize Active Directory data with the report database immediately. Depending on the Active Directory configuration and domain size, this operation can take awhile to complete.

   Or, alternatively, you can run the synchronization at a more convenient time, using the Report Services Control Panel.

3. Click **Finish** to close the configuration wizard.

   If the configuration was not successful, the configuration wizard provides some notes as to why the configuration failed. The notes may or may not include knowledge base articles that are available at the Delinea Technical Support web site.

Configuring a PostgreSQL Report Services Deployment

Follow these instructions if you're creating a new report services deployment using PostgreSQL or reconfiguring an existing PostgreSQL report services deployment.

To configure report services with a PostgreSQL database:

1. If you need to start the Delinea Report Services configuration wizard, go to the **Start** menu > **All Programs** > **Centrify Server Suite 2021.1** > **Report Services**, and choose **Configuration Wizard.**

   If you're continuing from the Delinea Management Services installer, the installer started the configuration wizard for you.

2. On the Welcome screen, click **Next** to continue.

3. If you have already set up report services, the Reconfiguring Report Services screen displays. Select **Reconfigure** and click **Next** to continue.

4. On the Database Type screen, select **PostgreSQL** and click **Next** to continue.

5. On the PostgreSQL screen, specify to create a new PostgreSQL installation or use an existing one.

   Because PostgreSQL doesn't have instances the way other databases do, Delinea recommends that you use an existing PostgreSQL database, if you already have one set up.

   For existing PostgreSQL installations, go to Step 8. Otherwise, for new installations, continue to Step 6.

6. To install a new PostgreSQL server, specify the PostgreSQL installer file location. You must specify a PostgreSQL installer version 11 or later. You can find the installer file in Common\PostgreSQL.

   Click **Next** to continue.

7. Specify the location of the PostgreSQL ODBC driver installer file. Delinea includes this file with the report services installer in Common\PostgreSQL.

   If you already have the official PostgreSQL ODBC drivers installed, this screen doesn't display.

   Click **Next** to continue.

8. Specify the PostgreSQL database settings:

   - **ODBC Driver**: For the PostgreSQL version that comes with report services, keep the default setting of PostgreSQL Unicode. This field can't be changed for new installations.
   - **Server**: If you're using an existing PostgreSQL server, enter the server name. For example, localhost or servername.acme.com.
   - **Port**: If you don't enter a port number, report services uses the default port 5432.
   - **Database**: This is the database name. The name can be up to 63 characters long, and the name cannot begin with an underscore (_) or dollar sign ($), and the instance name cannot contain any of the following special characters: a blank space, backslash (\), comma (,), colon (:), semi-colon (;), single quotation mark ('), ampersand (&), hyphen (-), number sign (#), or at sign (@).
   - **Database User**: This is your PostgreSQL administrator user. If you're using an existing PostgreSQL installation, the user must have the Create Database permission.

- **Password**: This is the password for your PostgreSQL administrator user. If you're using an existing PostgreSQL installation, this is the password for the user with the Create Database permission.
- **Confirm Password:** If you're creating a new installation, enter your password again to ensure the password is correct.
- **Additional Parameters**: Enter as needed. If you need to enter multiple characters, separate them with a colon (:).

  Note: The Configuration wizard verifies these settings after you've continued through all the configuration screens. Also, if you haven't installed the PostgreSQL ODBC driver, the Configuration wizard cannot verify these database settings.

9. If you selected domain-based reporting:
   1. In the Monitored Domain(s) screen, you can review and edit the list of domains that will be included for reporting. Add or remove domains as desired.

      For each domain, the configuration wizard lists the domain name and the domain controller name.
   2. Click **Next** to continue.
10. If you selected zone-based reporting and you use hierarchical zones:
    1. If you want data from all zones, select **Monitor all hierarchical zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
    2. Or, if you want to report data from specific zones, select **Monitor only specific hierarchical zones**.
    3. Click **Edit**.

       The Specify Forest for zone selection dialog box opens.
    4. Enter the forest name where the desired zones are and click **OK**.

       The Edit Monitored Hierarchical Zones dialog box opens.
    5. Enter the hierarchical zones by name, or expand the list of zones to locate the desired zones manually.
    6. If desired, specify to select the parent or child zones automatically.
    7. Select the zone by putting a checkmark in the box next to the zone name.
    8. When you're done specifying which hierarchical zones to monitor, click **OK** to close the dialog box and return to the Configuration wizard.

       Each zone that you've selected is listed in the Hierarchical Zones screen.
    9. Click **Next** to continue.
11. If you selected zone-based reporting and you use classic zones:
    1. If you want data from all zones, select **Monitor all classic zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
    2. Or, if you want to report data from specific zones, select **Monitor only specific classic zones**.
    3. Click **Edit**.

       The Specify Forest for zone selection dialog box opens.
    4. Enter the forest name where the desired zones are and click **OK**.

       The Edit Monitored Classic Zones dialog box opens.
    5. Select the classic zones to include in your reports. Select the zone by putting a checkmark in the box next to the zone name.

       You can filter the list of zones by entering a portion of the name and clicking **Filter**.
    6. When you're done specifying which classic zones to monitor, click **OK** to close the dialog box and return to the wizard. Click **Next** to continue.

12. For zone-based reporting, you can also specify which domain controller(s) that the report service connects to.

    If you don't specify which domain controller(s) to use, report services will use the default domain controller.

    1. Click **Add**.

       The Add Domain Controller dialog box opens.

    2. Enter the domain name and then select the domain controller from the list.
    3. Click **OK** to return to the Configuration wizard.

       The domain controllers that you selected are listed in the wizard screen.

    4. Click **Next** to continue.

13. In the Synchronization schedule screen, specify how often you want the reporting service to pull data from Active Directory.

    You can specify that the service synchronizes weekly, daily, every certain number of days, or every certain number of hours. The limit is 32,767 days or weeks.

    Click **Next** to continue.

14. Configure the user account that runs the service:

    1. In the Report Services options screen, specify the user account that will be used to run the service that synchronizes data from Active Directory and the reporting database.

       You can select a network service account, a managed service account, or another user account in Active Directory.

       You must specify a user account that has the required permissions. The configuration wizard verifies that the user has the correct level of access.

    2. Click **Next** to continue.
    3. The configuration wizard verifies that the specified user account has the required permission. An error displays if the permissions are inadequate.
    4. If the permission verification is successful, click **Close** to close the Verify permission window.

15. Review and complete the installation:

    1. In the Summary screen, review the installation details. If the installation settings are correct, click **Next** to continue.

       If you're installing a new database, it may take a few minutes.

    2. (Optional) In the completion screen, if the installation is successful, you can select the option to synchronize Active Directory data with the report database immediately. Depending on the Active Directory configuration and domain size, this operation can take awhile to complete.

       Or, alternatively, you can run the synchronization at a more convenient time, using the Report Services Control Panel.

    3. Click **Finish** to close the configuration wizard.

       If the configuration was not successful, the configuration wizard provides some notes as to why the configuration failed. The notes may or may not include knowledge base articles that are available at the Delinea Technical Support web site.

       Note: Delinea Report Services does not include a reporting solution for use with PostgreSQL.

Changing the Monitoring Mode for an Existing Report Services Deployment

You can easily switch from gathering report data based on domains or zones.

To change the monitoring mode for an existing report services deployment:

1. If you need to start the Delinea Report Services configuration wizard, go to the **Start** menu > **All Programs** > **Centrify Server Suite2021.1** > **Report Services**, and choose **Configuration Wizard.**

   If you're continuing from the Delinea Management Services installer, the installer started the configuration wizard for you.

2. On the Welcome screen, click **Next** to continue.
3. On the Reconfiguring Report Services screen, select **Switch the monitor mode** if you want to change whether report services uses domains or zones to synchronize Active Directory data. Click **Next** to continue.
4. On the Switch Monitor Mode screen, review the current and new mode settings. Click **Next** to continue.
    - To switch to domain-based reporting, go to Step 5.
    - To switch to zone-based reporting for hierarchical zones, go to Step 6.
    - To switch to zone-based reporting for classic zones, go to Step 7.
5. If you selected domain-based reporting:
    1. In the Monitored Domain(s) screen, you can review and edit the list of domains that will be included for reporting. Add or remove domains as desired.

        For each domain, the configuration wizard lists the domain name and the domain controller name.
    2. Click **Next** to continue.
6. If you selected zone-based reporting and you use hierarchical zones:
    1. If you want data from all zones, select **Monitor all hierarchical zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
    2. Or, if you want to report data from specific zones, select **Monitor only specific hierarchical zones**.
    3. Click **Edit**.

        The Specify Forest for zone selection dialog box opens.
    4. Enter the forest name where the desired zones are and click **OK**.

        The Edit Monitored Hierarchical Zones dialog box opens.
    5. Enter the hierarchical zones by name, or expand the list of zones to locate the desired zones manually.
    6. If desired, specify to select the parent or child zones automatically.
    7. Select the zone by putting a checkmark in the box next to the zone name.
    8. When you're done specifying which hierarchical zones to monitor, click **OK** to close the dialog box and return to the Configuration wizard.

        Each zone that you've selected is listed in the Hierarchical Zones screen.
    9. Click **Next** to continue.
7. If you selected zone-based reporting and you use classic zones:
    1. If you want data from all zones, select **Monitor all classic zones from forest(s)**. You can add or edit forests by clicking **Edit** and then adding or removing forests.
    2. Or, if you want to report data from specific zones, select **Monitor only specific classic zones**.
    3. Click **Edit**.

        The Specify Forest for zone selection dialog box opens.
    4. Enter the forest name where the desired zones are and click **OK**.

        The Edit Monitored Classic Zones dialog box opens.
    5. Select the classic zones to include in your reports. Select the zone by putting a checkmark in the box next to the zone name.

        You can filter the list of zones by entering a portion of the name and clicking **Filter**.

6. When you're done specifying which classic zones to monitor, click **OK** to close the dialog box and return to the wizard. Click **Next** to continue.

8. For zone-based reporting, you can also specify which domain controller(s) that the report service connects to.

   If you don't specify which domain controller(s) to use, report services will use the default domain controller.

   1. Click **Add**.

      The Add Domain Controller dialog box opens.

   2. Enter the domain name and then select the domain controller from the list.

   3. Click **OK** to return to the Configuration wizard.

      The domain controllers that you selected are listed in the wizard screen.

   4. Click **Next** to continue.

9. Configure the user account that runs the service:

   1. In the Report Services options screen, specify the user account that will be used to run the service that synchronizes data from Active Directory and the reporting database.

      You can select a network service account, a managed service account, or another user account in Active Directory.

      You must specify a user account that has the required permissions. The configuration wizard verifies that the user has the correct level of access.

   2. Click **Next** to continue.

   3. The configuration wizard verifies that the specified user account has the required permission. An error displays if the permissions are inadequate.

   4. If the permission verification is successful, click **Close** to close the Verify permission window.

10. Review and complete the installation:

    1. In the Summary screen, review the installation details. If the installation settings are correct, click **Next** to continue.

       If you're installing a new database, it may take a few minutes.

    2. (Optional) In the completion screen, if the installation is successful, you can select the option to synchronize Active Directory data with the report database immediately. Depending on the Active Directory configuration and domain size, this operation can take awhile to complete.

       Or, alternatively, you can run the synchronization at a more convenient time, using the Report Services Control Panel.

    3. Click **Finish** to close the configuration wizard.

       If the configuration was not successful, the configuration wizard provides some notes as to why the configuration failed. The notes may or may not include knowledge base articles that are available at the Delinea Technical Support web site.

Doing a Silent Install and Configuration

You can do what's called a silent install and configuration of Report Services, where you don't have to interact with a user interface and you plug in some configuration values ahead of time.

Doing a Silent Install of Report Services

Follow this procedure if you're installing Report Services components newly or upgrading to the latest version.

To silently install Report Services

1. Open a command window and run it as administrator.
2. Navigate to the Report Services folder in the installer package:

   Centrify-Server-Suite-**version**-mgmt-win64\DirectManage\Report Services

   where **version** refers to the Server Suite release version.
3. Run the following command to install the Report Services files:

   Centrify_RptServices-**version**-win64.msi /quiet

   where **version** refers to component version.

   The program installs the Report Services files into the default installation folder. You'll know that the install worked if you see newer files in that folder.

   You're now ready to configure Report Services. For new installations, see the next procedure. For upgrades, see Upgrading the reporting database silently.

Configuring a New Report Services Deployment Silently

There are a few main factors that drive which parameters to use in configuring Report Services silently (automatically without interaction):

- Are you creating a new database instance or publishing to an existing instance?
- Are you using PostreSQL or Microsoft SQL Server?
- Are you using Report Services in zone mode or domain mode?

When doing a silent configuration of report services, you create a file named config.json with the configuration parameters that you need for your deployment. For a list of the available parameters, see Report Services silent configuration parameters

Before you run the configuration program, be sure that you have the parameters set in the configuration file (config.json) and that you have the following ready to enter at the command line:

- ServiceAccount and ServiceAccountPassword
- PgSQLUser and PgSQLUserPassword (if using PostgreSQL)

The installer also supplies some sample configuration files (in the Report Services installation folder) that you can use as a guideline, depending on your deployment:

Microsoft SQL Server	config_sql_server_domain_mode_sample.json	config_sql_server_zone_mode_sample.json
PostgreSQL	config_postgresql_domain_mode_sample.json	config_postgresql_zone_mode_sample.json

To silently configure Report Services

1. Prepare the configuration file according to the settings you need.
2. Open a command line window as an administrator and run the report services configuration command according to the following usage:

```
Centrify.Report.Configuration.CLI.exe --ConfigFile "C:\\config.json" --ServiceAccount "Local System" --ServiceAccountPassword --PgSQLUser --PgSQLUserPassword
```

--ConfigFile	The configuration json file, including the path. You can name this file as desired, as long as it's a .json file.
--IsGroupManagedServiceAccount	Use this option to specify that the service account is a group managed service account (gmsa).
--ServiceAccount	The service account to use to configure report services.
--ServiceAccountPassword	The service account password. If you're using a built-in account such as "Local System", you don't have to specify the password.
--PgSQLUser	If you use PostrgreSQL as your database, specify the PostreSQL user name
--PgSQLUserPassword	The PostgreSQL password

The command does the configuration. If there is an issue or error, the command displays a message in red text. When the configuration succeeds, there's a message in green text indicating that the configuration is done.

Report Services Silent Configuration Parameters

Use the following parameters in the Report Services configuration file (config.json). You don't need to use all of them; it depends on which kind of database you use and whether you're configuring Report Services in domain mode or zone mode. These parameters match what you would enter in the installer interface.

The installer also supplies some sample configuration files (in the Report Services installation folder) that you can use as a guideline, depending on your deployment:

Microsoft SQL Server	config_sql_server_domain_mode_sample.json	config_sql_server_zone_mode_sample.json
PostgreSQL	config_postgresql_domain_mode_sample.json	config_postgresql_zone_mode_sample.json

AdditionalParam	string	(PostgreSQL only) Use this parameter if you have any additional PostgreSQL parameters that you need to specify.	
DBInstallerPath	string	If you're creating a new database instance, this parameter specifies the location of the database installer file.	"DBInstallerPath": "D:\\Common\\SQLEXPR\\SQLEXPRADV_x64_ENU.exe"
DBInstallationPath	string	If you're creating a new database instance, this parameter specifies where to install the new database instance. For directories or path separators, use \\ instead of \.	"DBInstallationPath": "C:\\Program Files\\Microsoft SQL Server\\130"
DBName	string	If you're creating a new database instance, the name of the new database.	"DBName": "Report"
DomainControllers	dictionary	(Zone mode only) -- key = domain, value = domain controller	{"test.com": "dc.test.com"}
		(Domain mode only) Use this parameter to specify	

DomainMode		domain mode. Inside of this parameter you specify the domains to synchronize to the reporting database.	"DomainMode": { "MonitoredDomains": { "test.com": "dc.test.com", "us.test.com": "dc.us.test.com" } }
ForestsForClassicZones	list of string values	(Zone mode only) The forest that contains the classic zones	
ForestsForHierarchicalZones	list of string values	(Zone mode only) The forest that contains the hierarchical zones	["companyA.com", "companyB.com"]
InstanceName	string	(SQL Server only) If you're creating a new database instance, the name of the new database instance. For directories or path separators, use \\ instead of \.	"InstanceName": "REPORTS"
IsDeployReport	true or false	(SQL Server only) Specifies whether or not to deploy reports.	"IsDeployReport": true
LogLevel	string	Specifies the amount of detail that the installer includes in the log file. The format of the parameter is as follows: Log level (Off|Critical|Error|Warning|Information|Verbose|Trace)	"LogLevel": "verbose"
MonitoredClassicZones (list)	list of string values	(Zone mode only) If you use classic zones, use this parameter to specify the list of zones to synchronize with report services.	["test.com/Program Data/Zones/ZoneC", "test.com/Program Data/Zones/ZoneD"]
MonitoredHierarchicalZones	list of string values	(Zone mode only) The list and location of hierarchical zones	["test.com/Program Data/Zones/hZoneA", "test.com/Program Data/Zones/hZoneB"]
NewDB	true or false	Specifies whether to create a new database instance or not	"NewDB": true
OdbcName	string	(PostgreSQL only) The name of the ODBC driver to use to connect with the PostgreSQL database.	"OdbcName": "PostgreSQL Unicode"
OdbcInstaller	string	(PostgreSQL only) The path and filename of the PostgreSQL installer file.	
Port	integer	(PostgreSQL only) The port for the PostgeSQL database. If you don't specify this parameter, the default port of 5432 is used.	"Port": "5432"
ReportManagerUrl	string	(SQL Server only) Specifies the report manager URL. You use this URL to edit, publish, and administer reports.	"ReportManagerUrl": "http://MYCOMPUTER/Reports_REPORTS"
ReportWebServiceUrl	string	(SQL Server only) Specifies the web service URL for deploying reports. You use the web service URL to read reports.	"ReportWebServiceUrl": "http://MYCOMPUTER/ReportServer_REPORTS"
		You use this parameter in conjunction with the ScheduleRule setting. With this parameter, you specify the number of hours, days, or weeks to configure how often the report synchronization will	"ScheduleRule": "interval", "ScheduleFrequency": 2, The

ScheduleFrequency	integer	happen. For example, if you set ScheduleRule to weekly and you specify ScheduleFrequency to 1, the synchronization will happen every week. If you change the ScheduleFrequency to 3, the synchronization will happen every 3 weeks.	above example specifies that the report synchronization will happen every 2 hours.
ScheduleRule	string	You use this parameter in conjunction with the ScheduleFrequency setting. With this parameter, you specify what sets of time to count when scheduling the frequency of report services synchronization. The options that you can specify are "daily", "interval", or "weekly". Interval specifies an hourly interval.	"ScheduleRule": "weekly", "ScheduleFrequency": 2, The above example specifies that the report synchronization will happen every 2 weeks.
ScheduleStartTime (string)	string	Specifies the time of day to start the report services synchronization, in the 24 hour format of hh:mm:ss.	"ScheduleStartTime": "14:00:00",
ScheduleWeekDays (list)	list of string values	Specifies on which days of the week the report services synchronization will happen. You can specify "all" for a daily synchronization or "none" to just do it manually as needed. Otherwise, you can specify one or more of the following for days of the week: "mon", "tue", "wed", "thu", "fri", "sat", "sun"	"ScheduleWeekDays": ["mon", "wed", "fri"]
Server	string	(PostgreSQL only) The name of the PostgreSQL server.	"Server":"localhost"
SqlCmdTimeout	int	SqlCommand timeout (second)	"SqlCmdTimeout": 3300
ZoneMode		(Zone mode only) Use this parameter to specify zone mode. Inside of this parameter you specify the MonitoredClassicZones and the MonitoredHierarchicalZones parameters and their respective settings.	ZoneMode": { "MonitoredClassicZones": [], "MonitoredHierarchicalZones": ["test.com/Program Data/Zones/hZoneA"], "ForestsForClassicZones": ["test.com"], "ForestsForHierarchicalZones": [], "DomainControllers": { "test.com": "dc.test.com", "us.test.com": "dc.us.test.com" } }

Upgrading from a Prior Version

You can install or upgrade the report services components using the Delinea Management Services installer and then use either the Report Services Configuration wizard or the Database Upgrade and Deployment wizard to get your database and reports set up. This table highlights which tools you can use, depending on whether you have a previous version of report services installed or not.

No	Install the report services components	Run the Configuration wizard to configure report services and deploy reports. For details, see Configuring Report Services and Deploying your Reports
Yes	Upgrade your report services components.	Run the Database Upgrade and Deployment wizard to upgrade your report database and deploy reports. For details, see Upgrading your report services database.

If you're upgrading from a version of Server Suite prior to 2016 or you don't currently have report services installed, you'll need to specifically indicate during the installation when you want to install the report services components - they aren't installed by default during an upgrade.

> **Note**: The Access Manager reports are still available, wherever you've installed Access Manager. Report services are in addition to the standard Access Manager reports.

Upgrading Your Report Services Database

If you're upgrading from a previous release of report services, you need to make sure that your report database is up to date. You'll also need to deploy your reports again so that they are based on the updated database.

The following SQL Server permissions are required in order to upgrade the report database with the Upgrade and Deployment wizard:

- Execute stored procedure permission on report database
- Create schema permission on report database
- Create table permission on report database
- Create view permission on report database
- Create stored procedure permission on report database
- Create type permission on report database
- Alter any schema permission on report database
- Insert, Delete, Update, Select and Execute permissions on the schema "Dbo", "RawData", "ReportData", "ReportView" and "ConfigData" on report database

In order to deploy reports, you must have the Microsoft SQL Server Reporting Services role of Content Manager. For details for how to grant SSRS roles, see Granting access in SSRS to reports.

To upgrade your report database:

1. From the Start menu, locate and run the **Delinea Report Services Upgrade and Deployment wizard**.
2. In the initial screen, click **Next** to continue.
3. The wizard upgrades the database automatically.

   The database upgrade changes are saved to the database after you exit the wizard later.
4. If you have deployed reports before, configure where to back up the existing reports and where the new reports will be deployed.

   If you haven't deployed reports before, you're prompted to specify where to deploy reports.

   If desired, you can select the option to not backup nor deploy reports.

   Click **Next** to continue.
5. In the Summary screen, review the settings and if they're correct, click **Next** to continue.

The wizard upgrades your report database.

6. In the completion screen, click **Finish** to exit the wizard.

   (If the upgrade failed for any reason, the Summary screen displays some details about why the upgrade failed.)

   Your report database is updated and your reports are deployed, if you specified the option to do so.

   Note: After upgrade, you should perform a full synchronization before an incremental update is allowed. (Ref: CS-40029a)

Upgrading from Versions Before 2016

As of Server Suite 2016 the report services feature provides reports. If you're upgrading from a version prior to release 2016 and you're accustomed to the Access Manager reports, this section covers the differences between the reports.

If you want to know which Delinea report services reports correspond to the Access Manager reports, below is a list. The reports are listed according to the Access Manager report so that you can easily determine which new report you want to use instead.

Classic Zone Access Manager Reports

These Classic Zone reports correspond to the report services reports as follows:

Classic Zone - Authorization Report for Computers	Lists each computer in the zone and indicates which users are allowed to access each computer.	Authorization Report
Classic Zone - Authorization Report for Users	Lists each user account in the zone and indicates which computers each user can access.	
Classic Zone - User Privileged Command Rights Grouped by Zone	Lists the privileged commands that each user has permission to run and the scope to which the user's rights apply.	Classic Zone - User Privileged Command Rights Report
Classic Zone - User Role Assignments Grouped by Zone	Lists the role assignments for each user in each zone.	Classic Zone - User Role Assignment Report
Classic Zone - Users Report	Lists information from the UNIX profile for each user in each classic zone.	
Classic Zone - Zone Role Privileges	Lists the roles that are defined for each classic zone and the rights granted by each of these roles.	Zone Role Privileges Report

Hierarchical Zone Access Manager Reports

These Hierarchical Zone reports correspond to the report services reports as follows:

Hierarchical Zone - Computer Effective Audit Level	Lists the audit level in effect for computers in each zone.	Hierarchical Zone - Effective Audit Level
Hierarchical Zone - Computer Effective Rights	Lists the privileges granted on each computer.	Hierarchical Zone - Effective Rights Report
Hierarchical Zone - UNIX User Effective Rights	Lists the effective rights for each UNIX user on each computer. The report shows the name of the right, it's type, and where it is defined.	
Hierarchical Zone - Windows User	Lists the effective rights for each Windows user on each computer. The report	

Effective Rights	shows the name of the right, it's type, and where it is defined.	
Hierarchical Zone - Computer Effective Roles	Lists the roles assigned on each computer.	Hierarchical Zone - Effective Role Report
Hierarchical Zone - Computer Role Assignments	Lists the computer roles that are defined for each zone. The report includes the users and groups and their associated roles.	Hierarchical Zone - Computer Role Assignments Report
Hierarchical Zone - Computer Role Membership	Lists the computer roles that are defined for each computer and the zone to which they belong.	Hierarchical Zone - Computer role Membership Report
Hierarchical Zone - Computer Role Membership Grouped by Zone	Lists the computer roles that are defined for each computer grouped by the zone to which they belong.	

All Zone Access Manager Reports

These reports correspond to report services reports as follows:

Computer Summary Report	Lists computer account information for each computer in each zone.	Computers Summary Report
Computers Report	Lists computer account information for each computer in each zone.	
Groups Report	Lists group information for each group in each zone.	Groups Report
Stale Computers Report	Lists the stale computers.	Stale Computers Report
User Accounts Report	Lists account details for the users that have UNIX profiles in each zone. The report includes the Active Directory display name, the Active Directory login name, the Active Directory domain for the account, and details about the account status, such as whether the account is configured to expire, locked out, or disabled and the date and time of the account's last login.	User Accounts Report
Zones Report	Lists the zone properties for each zone. The report includes the zone name, list of available shells, the default shell, the default home directory path, the default primary group, the next available UID, reserved UIDs, the next available GID, and reserved GIDs.	Zones Report

Reports that are New to Access Manager Report Users

In addition to converting the content of the Access Manager reports into the report services reports, there are also the following new reports:

- Hierarchical Zone - Computer Role Effective Assignments Report (one for UNIX, one for Windows)
- Hierarchical Zone - Zone Effective Assignments Report (one for UNIX, one for Windows)
- Attestation reports for SOX and PCI compliance

Upgrading the Reporting Database Silently

If desired, you can upgrade your report services database without any user interaction, after you install the latest version of the Report Services components. You supply any of the parameters in the table below when you run the command line program. These parameters match the settings in the Upgrade and Deployment wizard.

ReDeployReport	switch	Specifies whether or not to redeploy reports after you upgrade the database. If you include this parameter, the service will redeploy reports. If you don't include this parameter, the service doesn't redeploy reports.	--ReDeployReport
ReportBackupFolder	string	Specifies the path and folder location to backup existing reports before upgrading. This option only applies if you specify ReDeployReport to yes. If you don't specify a value for this property, the service uses the default value of "Backup reports".	
WebServiceURL	string	Specifies the web service URL for deploying reports. You use the web service URL to read reports. If you don't specify this value, the service uses the current setting.	"ReportWebServiceUrl": "http://server1/ReportServer_REPORTS"
ReportManagerURL	string	Specifies the report manager URL. You use this URL to edit, publish, and administer reports. If you don't specify this value, the service uses the current setting.	"ReportManagerUrl": "http://server1/Reports_REPORTS"

To silently upgrade Report Services:

1. Install the latest version of the Report Services components. For details, see Silently installing Report Services.
2. At the command line (be sure to run as administrator), run the following command with the desired upgrade parameters, as listed in the table above. None of the parameters are mandatory.

   ```
   Centrify.Report.Upgrade.Cli.exe --ReDeployReport --ReportBackupFolder "previous-reports"
   ```

The upgrade program lists out each upgrade task that it performs as it progresses. When the program finishes, there's a message that says the upgrade is finished.

Administering Report Services with the Report Control Panel

You can use the Delinea Report Services Control Panel for the following tasks:

General Tab

- View the status of data synchronization from Active Directory to the report database
- View the domains or zones that are included for reporting
- Start, stop, or restart the reporting service.

Monitored Zones Tab

> **Note**: This tab appears only if you've configured reporting based on specific zones instead of domains.

- Edit the Hierarchical or Classic zones that you want to include in your reports. You can add or remove zones, as desired, and you can select zones from other trusted forests.

Settings Tab

- Configure when the reporting service synchronizes data from Active Directory to the reporting database
- Change the user account that runs the reporting service.
- Add, edit, or remove domain controllers (in zone-based monitor mode) or domains (in domain-based monitor mode).
- If you're using a PostgreSQL database, you can test the connection to the database.

Troubleshooting Tab

- View the log files and set the level of detail that are collected in the log files.
- Export diagnostics data for use by Delinea Technical Support (if technical support requests that you do so).
- Rebuild or refresh the reports data
- Validate that the reporting service has the correct permissions to read data from the monitored domains and replicate the data.

Configuring SQL Server Reporting Services (SSRS)

This section includes the following topics:

- Adding your report services web site to your Internet Explorer trusted sites
- Granting access in SSRS to reports
- Providing reports to your users or auditors
- Sharing reports by email or file sharing with report subscriptions

Adding Your Report Services Web Site to Your Internet Explorer Trusted Sites

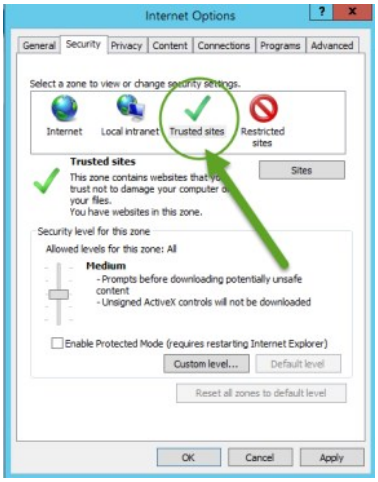
Chrome, Firefox, and Safari are NOT supported for SSRS. This is a Microsoft limitation.

In order to view the reports in Internet Explorer, you also have to add the report server as a trusted site. (If you're running an evaluation version, you can also choose to disable the Internet Enhanced Security configuration, but it's not recommended to do so.)

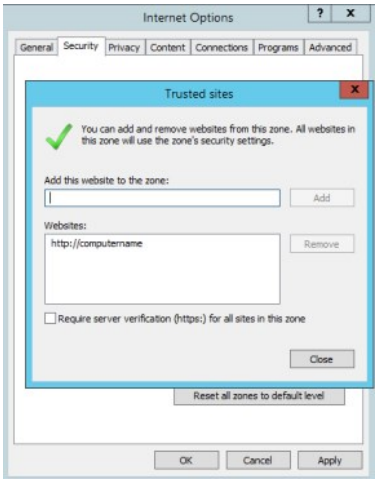
Please consult Microsoft documentation for the most current instructions for Internet Explorer configuration. However, for your convenience, here's a quick reminder of how to add a trusted site.

To configure Internet Explorer to trust the report services deployment site in the local intranet zone:

1. In Internet Explorer, go to **Tools** > **Internet Options**.
2. Click **Security**.
3. In the Zones area, click **Trusted Sites**.



4. Click **Sites**.
5. In the Trusted Sites dialog box, enter the web site address for your report services deployment, and click **Add**.



For example, enter a URL that looks something like this: http://computername/reportinstancename.

6. Click **Close**, and then click **OK** to save the changes.

Granting Access in SSRS to Reports

Before you provide reports to your users, you need to give them the appropriate access within the Microsoft SQL Server Reporting Services application. You use the SSRS role-based security to assign Active Directory users and groups to SSRS roles for both the site and folders.

Anyone reading reports will also need to configure their Internet Explorer installation, as mentioned in Adding your report services web site to your Internet Explorer trusted sites.

Please consult Microsoft documentation for the most current instructions for security configuration and granting access in SSRS. For example, some information can be found at this link:

https://docs.microsoft.com/en-us/sql/reporting-services/report-server/configure-a-native-mode-report-server-for-local-administration-ssrs?view=sql-server-2016

However, for your convenience, a couple procedures are below.

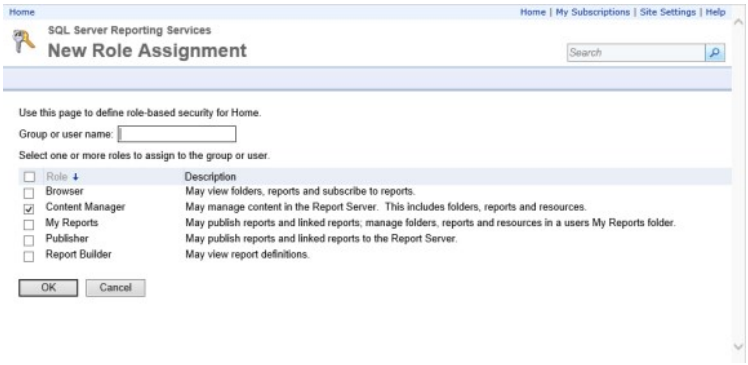
To grant report **administrator** access in SSRS (SQL Server Reporting Services):

1. Run Internet Explorer as Administrator.
2. In Internet Explorer, go to your Report Manager URL.

   You can open the Microsoft Reporting Services Configuration Manager to view the Report Manager URL.

   Internet Explorer opens SQL Server Reporting Servies to your Report Manager URL.
3. Click **Site Settings**, and create a new role assignment so that you can assign the desired Active Directory group to the "System Administrator" role in SSRS.

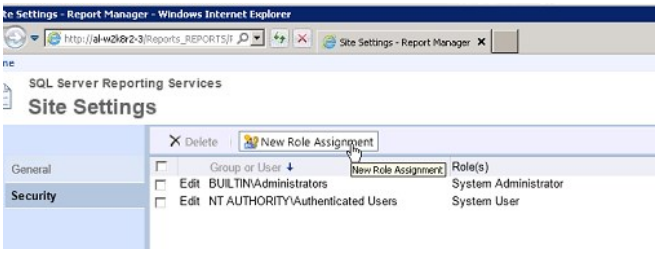
   To create a new role assignment, click **Security**, then **New Role Assignment**.
4. Enter the group or user name (in the domain\username formet), select **System Administrator**, and click **OK**.
5. Click **Home**, and then click **Folder settings**. From there, create a new role assignment so that you can grant access to the "Content Manager" role.



6. To grant access so that the user can **edit or build** reports, you can give them additional permissions in SSRS, such as the Report Builder permission to the Home folder.

To grant report **read** access in SSRS (an overview):

1. In SSRS, go to Site Settings, and create a new role assignment so that you can assign the desired Active Directory group to the "System user" role in SSRS.



   By default, all authenticated users are assigned to the System User role.

2. In SSRS, go to the Home folder, and then click Folder settings. From there, create a new role assignment so that you can grant access to at least the "Browser" role.

3. To grant access so that the user can **edit or build** reports, you can give them additional permissions in SSRS, such as the Report Builder permission to the Home folder.

Providing Reports to Your Users or Auditors

After you've made sure that your users have the appropriate read access to reports within SSRS, you provide the report URL to your users and instruct them to access that URL within your domain and using the Internet Explorer browser. They may also need to add the report URLs to their trusted domains list; for details, see Adding your report services web site to your Internet Explorer trusted sites.

Sharing Reports by Email or File Sharing with Report Subscriptions

You can also create report subscriptions so that you can easily share reports by way of email or a file share. These are features of Microsoft SSRS, and the Microsoft documentation has the latest information.

In order to share reports by email, you first need to configure your report server for email delivery. For details, see https://msdn.microsoft.com/en-us/library/ms345234(v=sql.110).aspx.

For details for how to share reports by email or file sharing, see https://msdn.microsoft.com/en-us/library/ms189680(v=sql.110).aspx.

Re-Deploying the SQL Server Reports to SSRS

You can re-deploy your reports without needing to go through the entire Delinea Report Services configuration wizard. You can only re-deploy reports if you use SQL Server for your report database.

To configure Delinea report services using the configuration wizard:

1. If you need to start the Delinea Report Services configuration wizard, go to the **Start** menu › **All Programs** › **Centrify Server Suite 2021.1** › **Report Services**, and choose **Configuration Wizard.**

   If you're continuing from the Delinea Management Services installer, the installer started the configuration wizard for you.

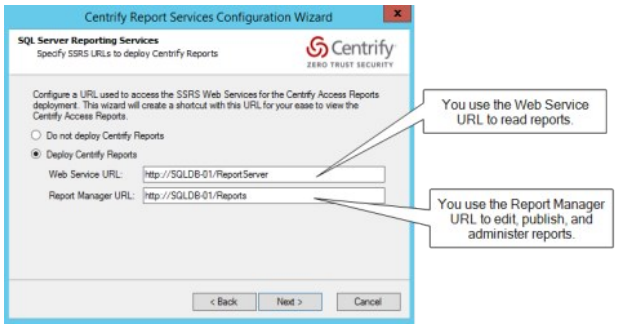
2. On the Welcome screen, click **Next** to continue.

3. On the Reconfiguring Report Services screen, select **Deploy reports only** and click **Next** to continue.

4. Deploy the reports:

   1. In the SQL Server Reporting Services screen, specify whether to deploy the Server Suite reports (or not).

      If you plan to use a reporting solution other than Microsoft SQL Server Reporting Services, do not deploy the reports.

      This screen also lists the URLs for the Reporting Web Service and Report Manager. You'll use these URLs later to access to the reports.

      If you're using a production server of SQL Server and SSRS, you can configure them to use HTTPS. For details, see Microsoft SQL Server and SSRS documentation, such as https://msdn.microsoft.com/en-us/library/ms345223.aspx.

      The configuration wizard populates the report URLs automatically. If you had specified to use an existing SQL Server instance, the configuration wizard retrieves the existing web service URL and report manager URL for your SQL Server instance.

      For an existing SQL Server instance, you can open the Microsoft Reporting Services Configuration Manager to view the Web Service and Report Manager URLs.

   2. Click **Next** to continue.

      Review and complete the installation:

   3. In the Summary screen, review the installation details. If the installation settings are correct, click **Next** to continue.

      If you're installing a new database, it may take a few minutes.

   4. (Optional) In the completion screen, if the installation is successful, you can select the option to synchronize Active Directory data with the report database immediately. Depending on the Active Directory configuration and domain size, this operation can take awhile to complete.

      Or, alternatively, you can run the synchronization at a more convenient time, using the Report Services Control Panel.

   5. Click **Finish** to close the configuration wizard.

      If the configuration was not successful, the configuration wizard provides some notes as to why the configuration failed. The notes may or may not include knowledge base articles that are available at the Delinea Technical Support web site.

Viewing Default Reports

This section covers how to open a report, and provides some basic information on each of the default reports.

Opening a Report

You open a report by going to the report folder URL in Internet Explorer. Click a report to open it.

In general, you and your users access the reports from a URL. The URL has a format like this:

"http://hostname/Reports_reportDBname"

Filtering Report Data by Zone

When you view a report, you can filter the report data by zone. In the zone drop-down filter, report services lists each zone by its full zone hierarchy, so that you can choose based on parent or child zones. For example, if you have a child zone named California as part of a parent zone West which is part of the parent zone United States, the zone appears in the list as "United States/West/California.

Zones are listed in the zone drop-down filter in alphabetical order, and the first zone in the list is the default zone. When you first open a report, report services initially generates the report data based on the default zone.

Default Access Manager Reports

Report Services Reports: Not Specific to Classic or Hierarchical Zones

Authorization report	This report lists each computer or user account, and which users are allowed to access each computer.	Access Level Computer domain Computer Name User domain User name User Type Zone Zone domain
Computers Summary report	Lists computer account information for each computer in each zone.	Computer domain Computer name Platform Zone Zone domain Zone type
Delegation report	Lists which users, groups, computers, group managed service accounts (gMSA), managed service accounts (MSA), and which well-known SIDs have which delegation tasks.	Delegation Task Target Target Domain Target Name Trustee Trustee Domain Trustee Type Zone
Effective delegation report	Lists which Active Directory users, Active Directory groups, group managed service accounts (gMSA), and managed service accounts (MSA) have which delegation tasks.	Active Directory User Domain Active Directory User Name Delegation Task Target Target Domain Target Name Zone
Groups report	Lists group information for each group in each zone, including the Active Directory group name, the UNIX group name, the UNIX group identifier (GID), and whether the group is an orphan. If the group is for local users, the local group status indicates whether the group is enabled or disabled for local access.	Active Directory Group name Active Directory Group domain Group Type Is Orphan Local Group Status UNIX Group Name Zone Zone Domain Zone Type
Stale Computers report	Lists the stale computers. Stale computers are those where the password hasn't changed for 90 or more days.	Computer Domain Computer Name Zone Zone domain
User Accounts Report	Lists account details for Active Directory users who are related to each zone. The report includes the Active Directory display name, the Active Directory login name, the Active Directory domain for the account, and details about the account status, such as whether the account is configured to expire, locked out, or disabled and the date and time of the account's last login.	Active Directory user name Domain Enabled

Users Report	Lists user information for each user in each zone. If the user is a local user, the local user status indicates whether the user is enabled or disabled for local access.	Active Directory user Active Directory user domain UNIX name Enabled Is Orphan Local User Status User Type Zone Zone domain Zone type
Zone Role Privileges Report	Lists the roles that are defined for each hierarchical zone and the rights granted by each of these roles.	Right name Right type Role name Zone Zone domain Zone type
Zones Report	Lists the administrative tasks and properties for each zone and the users or groups have been delegated to perform each task. This report indicates which users or groups have permission to perform specific tasks, such as add groups, join computers to a zone, or change zone properties.	Zone Zone domain

Delinea Report Services Reports: Classic Zone Reports

Classic Zone - User Privileged Command Rights Report	Lists the privileged commands that each user has permission to run and the scope to which the user's rights apply.	Classic zone Privileged command name User name Zone domain
Classic Zone - User Role Assignment Report	Lists information from the UNIX profile for each user in each classic zone. Lists the role assignments for each user in each zone. The report includes the domain name, user profile name, the list of roles the user is assigned to in each zone, and the scope to which the user's role assignment applies.	Classic zone Role User domain User name Zone domain

Delinea Report Services Reports: Hierarchical Zone Reports

Hierarchical zone - Computer Role Assignments Report	Lists the computer roles that are defined for each zone. The report includes the users and groups and their associated roles.	Role name Computer Role name Zone Zone domain
Hierarchical zone - Computer Role Effective Assignments Report	Lists the roles assigned on each computer. There are separate reports for UNIX and Windows computers.	Computer role Right Right type Role User Domain User Name Zone Zone Domain
Hierarchical Zone - Computer Role Membership Report	Lists the computer roles that are defined for each computer and the zone to which they belong.	Computer Domain Computer Name Computer Role in Zone Computer Role Name Join To Zone Domain
Hierarchical Zone - Effective Audit Level Report	Lists the audit level in effect for computers in each zone.	computer domain computer name User domain user name zone zone domain
Hierarchical Zone - Effective Rights Report	Lists the privileges granted on each computer and the effective rights for each Windows and UNIX user on each computer.	computer domain computer name Right Right type Role User domain user name zone zone domain

Hierarchical Zone - Effective Role Report	Lists the role assignment on each computer in the zone.	computer domain computer name Role User domain user name zone zone domain
Hierarchical Zone - Users Report	Lists the users and the computers to which they have access in the zone. If the user is a local user, the local user status indicates whether the user is enabled or disabled for local access.	Active Directory user Active Directory user domain Computer Computer domain Is orphan Is secondary Local User Status UNIX name User type Zone Zone domain
Hierarchical Zone - Zone Effective Assignments Report	Lists the roles that are defined for each hierarchical zone and the rights granted by each of these roles, including where each right is defined. There are separate reports for UNIX and Windows users.	Right Right type Role User domain user name zone zone domain

Default SOX Attestation Reports

To help your department comply with Sarbanes-Oxley audit requirements, Delinea provides some default SOX reports. These reports show you who has access to computers, what roles and rights users have, and similar data that's needed to show SOX compliance.

SOX reports provide the following kinds of information:

- **Computers**: Who has access to these computers, what are the roles, rights, and groups that they belong to
- **Groups**: Which users are in which groups, what are the roles, rights, and what computers can these users access
- **Users**: What their role assignments are, what rights the users have, which groups they belong to, and which computers they have access to
- **Roles**: Which computers the rules have access to, what rights are assigned to the group, and which groups are assigned to which roles

You can find the SOX reports in SSRS by going to the Centrify Report Services > Attestation > SOX reports folder.

> **Note**: In larger environments, you can save processing time when running an attestation report (PCI or SOX report) by choosing to exclude the chart from the report. When you open the report, select **True** for the **Exclude chart for faster report generation** option.

For a description of how report services calculates the data for the charts in the SOX reports, see How objects are counted for the PCI and SOX report charts.

Here is a list of the SOX reports, along with a brief description and how you can filter the results.

SOX - Login Report - By Computer	For each computer, this report displays the users who can log in. For each user who can log in, the report shows the role, assignment location, and assignee.	Computer Computer group Computer role Zone Zone Domain Zone Type
SOX - Login Report - By Group	For each Active Directory group, this report lists the computers and role assignment information.	Active Directory group Zone Zone Domain Zone Type
SOX - Login Report - By Role	For each role, this report lists the computers assigned to that role.	Role Zone Zone Domain Zone Type
SOX - Login Report - By User	For each user, this report lists the computers that the user can access as well as the role assignment information.	User Zone Zone Domain Zone Type
SOX - Login Summary Report	This report provides a summary of who can log in to which computer.	Computer Computer group Computer role Local User Status User User group User type Zone domain Zone type Zone
SOX - Rights Report - By Computer	For each computer, this report lists the users who have which login and other privileges and what the role assignments are.	Computer Computer Group Computer role Right type Zone Zone Domain Zone Type

SOX - Rights Report - By Group	For each Active Directory group, this report lists the computers have which login and other privileges and what the role assignments are.	Active Directory group Right type Zone Zone Domain Zone Type
SOX - Rights Report - By Role	For each role, this report lists the computer and rights available on that computer.	Role Zone Zone Domain Zone Type
SOX - Rights Report - By User	For each user, this report lists the Active Directory group, computers, and role assignment.	Right type User Zone Zone Domain Zone Type
SOX - Rights Summary Report	This report provides a summary of which rights are granted to which users on which computers.	Computer Computer group Computer role Local User Status Right type User group User User type Zone Zone Domain Zone type

Note: When you view the collection of reports in Internet Explorer, you may also see some sub-reports listed. These are not actual reports but views that support the actual reports; due to a limitation with Microsoft SSRS, these sub-reports may display even though they're not meant to be used. Please do not click any reports that have names that begin with SubReport.

Note: In these reports, Computer Role and Computer Group filters return records assigned to those roles or groups but not where the role assignment is defined. For example, if you filter records for Zone1_CompRoleA, the report lists all computers that are in the computer role named Zone1_CompRoleA.

Note: The charts in the PCI & SOX reports do not consider role assignments that are granted to "All Active Directory Users," and the reports only consider role assignments that are granted to specific users and groups when counting computer access and privileges. On the other hand, the detailed report shows all the login and privilege information from all role assignments (including those that are granted to "All Active Directory Users").

Default PCI Attestation Reports

To help your department comply with PCI audit requirements, Delinea provides some default PCI attestation reports. These reports show you who has access to computers, what roles and rights users have, and similar data that's needed to show PCI compliance.

PCI reports provide the following kinds of information:

- Computers: Which users have access to these computers, what are their roles and rights
- Groups: Which users are in which groups, what are their roles and rights, and which computers do they have access to
- Users: What role is the user assigned to, what rights does the user have, and which computers does the user have access to
- Roles: What computers do these roles have access to and what rights do they have

You can find the PCI reports in SSRS by going to the Centrify Report Services > Attestation > PCI reports folder.

Note: In larger environments, you can save processing time when running an attestation report (PCI or SOX report) by choosing to exclude the chart from the report. When you open the report, select **True** for the **Exclude chart for faster report generation** option.

For a description of how report services calculates the data for the charts in the PCI reports, see How objects are counted for the PCI and SOX report charts.

Here is a list of the PCI reports, along with a brief description and how you can filter the results.

PCI - Login Report - By Computer	For each computer, this report displays the users who can log in. For each user who can log in, the report shows the role, assignment location, and assignee.	Computer Computer group Computer role Zone Zone Domain Zone Type
PCI - Login Report - By Group	For each Active Directory group, this report lists the computers and role assignment information.	Active Directory group Zone Zone Domain Zone Type

PCI - Login Report - By Role	For each role, this report lists the computers assigned to that role.	Role Zone Zone Domain Zone Type
PCI - Login Report - By User	For each user, this report lists the computers that the user can access as well as the role assignment information.	User Zone Zone Domain Zone Type
PCI - Login Summary Report	This report provides a summary of who can log in to which computer.	Computer Computer group Computer role Local User Status User User group User type Zone domain Zone type Zone
PCI- Rights Report - By Computer	For each computer, this report lists the users who have which login and other privileges and what the role assignments are.	Computer Computer Group Computer role Right type Zone Zone Domain Zone Type
PCI- Rights Report - By Group	For each Active Directory group, this report lists the computers have which login and other privileges and what the role assignments are.	Active Directory group Right type Zone Zone Domain Zone Type
PCI- Rights Report - By Role	For each role, this report lists the computer and rights available on that computer.	Role Zone Zone Domain Zone Type
PCI- Rights Report - By User	For each user, this report lists the Active Directory group, computers, and role assignment.	Right type User Zone Zone Domain Zone Type
PCI - Rights Summary Report	This report provides a summary of which rights are granted to which users on which computers.	Computer Computer group Computer role Local User Status Right type User group User User type Zone Zone Domain Zone type

Note: When you view the collection of reports in Internet Explorer, you may also see some sub-reports listed. These are not actual reports but views that support the actual reports; due to a limitation with Microsoft SSRS, these sub-reports may display even though they're not meant to be used. Please do not click any reports that have names that begin with SubReport.

Note: In these reports, Computer Role and Computer Group filters return records assigned to those roles or groups but not where the role assignment is defined. For example, if you filter records for Zone1_CompRoleA, the report lists all computers that are in the computer role named Zone1_CompRoleA.

Note: The charts in the PCI & SOX reports do not consider role assignments that are granted to "All Active Directory Users," and the reports only consider role assignments that are granted to specific users and groups when counting computer access and privileges. On the other hand, the detailed report shows all the login and privilege information from all role assignments (including those that are granted to "All Active Directory Users").

How Objects are Counted for the PCI and SOX Report Charts

This section describes how objects are counted for the charts that you see in the PCI & SOX reports.

Login Report Charts

In login reports, we count how many computers each user can log in to, how many users can log in to each computer, and how many roles are granted with login rights.

In hierarchical zones, a role is considered to be granted with a login right if one or more of the following rights are granted to the role:

- Console login is allowed
- Remote login is allowed
- Password login and non-password login are allowed
- Non password login is allowed

In classic zones, a role is considered to be granted with a login right if at least one PAM right is granted to the role.

In the graphs that report the number of users who can log in to a computer, or the number of computers that a user is logged in to; the graphs only consider effective users. An effective user is one who has a complete user profile in a classic zone. In hierarchical zones, an effective user must also have been granted the login right through any role that is assigned to users/groups. Note that a "login right" obtained from a role that is assigned to "All AD users" is not considered in the graphs.

A local user is counted as an effective user in hierarchical zones if the user is granted the "User is visible" right from any effective role assignment.

Login Report – By Computer charts

Computers with Most Access chart

This chart ranks the computers by the number of effective users and shows the top 10 computers.

User Roles Count for Computers with Most Access chart

This chart ranks the computers by the number of roles that assign login rights to users or groups on the computer.

Users with Most Access chart

This chart ranks the users by the number of computers that each one can log in to, and shows the top 10 users.

Login Report – By Group charts

Roles with Most Access chart (by Group)

This chart ranks all the roles that are assigned to any group by the number of computers that the role grants login access to (regardless of how many groups are assigned to each role), and shows the top 10 roles.

Groups with Most Members chart

This chart shows the top 10 groups that have most members, including those from nested groups.

Login Report – By Role charts

Roles with Most Access chart (by Role)

This chart ranks all the roles that are assigned by the number of computers that the role grants login access to, and shows the top 10 roles.

Roles with Most Users chart

This chart ranks the number of users for which each role is effective (regardless of the role assignment scope), and shows the top 10 roles.

Roles with Most Rights chart

This chart ranks the assigned roles (regardless of the role assignment scope) with login rights by the number of granted privilege access rights.

Login Report – By User charts

Users with Most Access On Computers chart

This chart ranks the users by the number of computers that each one can log into, and shows the top 10 users.

Login Roles Count for Users with Most Access On Computers chart

This chart ranks the users by the number of effective roles that grant login access to any computer, and shows the top 10 users.

Login Summary Report charts

Computers With Most Access chart

This chart ranks the computers by the number of effective users and shows the top 10 computers. Both Active Directory and local effective users are considered.

Users With Most Access chart

This chart ranks all effective users by the number of computers that each user can log into, and shows the top 10 users.

Rights Report Charts

In each rights report, the privileged access right enables the user to create additional working environments or to run specified applications with different privileges. The following five privileged access rights are included in rights reports.

- Network Access right
- Desktop right
- Application right
- Commands
- Use restricted environment

Each privileged access right is counted in the reports only when the role with one of these rights is assigned to users and/or groups. However, the privileged right granted using 'All AD user' is not counted.

Rights Report – By Computer Charts

Computers with Most Privileged Access chart

This chart ranks the computer by the number of distinct privileged access rights that are effective on each computer, and shows the top 10 computers. A privileged access right is counted as one regardless of the number of users or roles that is granted or assigned the right in the computer.

Computer Roles with Most Privileged Access Chart

This chart ranks all the computer roles by the number of distinct privileged access rights assigned to each computer role, and shows the top 10 computer roles.

Privileged Access with Most Computers Chart

This chart ranks all privileged access rights by the number of computers that each right is effective on, and shows the top 10 rights.

Rights Report – By Group Charts

Groups with Most Privileged Access Chart

This chart ranks the group by the number of distinct privilege access rights granted to each group, and shows the top 10 groups. The privilege access rights are evaluated based on all roles that are assigned to groups, regardless of the scope of the assignments.

Rights Report – By Role Charts

Computer Roles with Most Privileged Access Chart

This chart ranks all the computer roles by the number of distinct privileged access rights assigned to each computer role, and shows the top 10 computer roles.

User Roles with Most Privileged Access Chart

This chart ranks the assigned roles (regardless of the role assignment scope) with login rights by the number of granted privileged access rights.

Rights Report – By User Charts

Users with Most Privileged Access Chart

This chart ranks all users by the number of distinct privileged access rights granted (regardless of the number of computers) and shows the top 10 users.

Computer Role Count for Users with Most Privileged Access Chart

This chart ranks all users by the number of distinct privilege access rights granted. For the top 10 users, it shows the number of computer roles where the user is assigned to any role in that computer role.

Rights Summary Report Charts

Computers with Most Privileged Access Chart

This chart ranks the computer by the number of distinct privileged access rights that are effective on each computer, and shows the top 10 computers. A privileged access right is counted as one regardless of the number of users or roles that is granted or assigned the right in the computer.

Users with Most Privileged Access Chart

This chart ranks all effective users by the number of distinct privileged access rights granted (regardless of the number of computers) and shows the top 10 users.

Most Dominant Privileges on Computers chart

This chart ranks all privileged access rights by the number of computers that each right is effective on and shows the top 10 rights. The number of users where the right is effective in each computer is not considered in the ranking.

Building Custom Reports

You can build your own reports with data from the Delinea report services database by using your own reporting tool or Microsoft SQL Server Reporting Services.

Requirements and Recommendations

In order to build your own reports or customize existing reports, you also need to have the SSRS Report Builder installed where you have SSRS installed.

Known Limitations and Recommendations

- Use the same domain where Microsoft SSRS is installed. If you try to use SSRS in a domain that is different from the domain where SSRS is installed, you may have some difficulty accessing reports. For example, if your computer runs in the acme.com domain and you have SSRS installed in a test domain of wiley.coyote.com, you may run into issues accessing the reports.
- If you're accessing SSRS from a different domain, make sure that you enter your credentials and save them.
- When you log in to SSRS, make sure that the user you're logging in as has at least the system user role, and at least read access to the folder (according to the folder settings in SSRS).

An Overview of Report Building Tasks

Microsoft documentation contains specific instructions for how to create custom reports using SSRS Report Builder. Included here is the overall process; please consult Microsoft SSRS Report Builder documentation for details.

For example, here's a link to Microsoft information on using SQL Server Reporting Services 2012: https://technet.microsoft.com/en-us/library/hh338693.aspx.

An overview of how to build custom reports using SSRS and Delinea report services data:

1. Open Internet Explorer to the deployed reports URL.
    - Make sure that you have the correct access permissions in SSRS for building reports. For details, see Granting Access in SSRS to Reports.
    - It's recommended that you log in to the deployed reports URL as a user with Report Building permissions, but not database administrator permissions. If you log in as a user with access to all tables in the reporting database, you may see tables that you cannot use in custom reports. Delinea exposes the views for you to use in your custom reports.
2. Open Microsoft SQL Server Report Builder, and create the dataset that connects you to the reporting data source.

    (The dataset is the set of data retrieved from the database, and the data source is the connection information for the database.)
3. Create a new report that's based on the data set that you just created.
4. Design a query using the provided views.
5. Run the report to make sure that you get data in the report.
6. Edit the report as desired.
7. Save the report.

    Microsoft SSRS saves the report as a .RDL file.
8. Publish the report by publishing the RDL file.

Migrating Custom Reports from SQL Server Express

If you create custom reports using the included version of SQL Server 2008 R2 Express edition, you can migrate those custom reports over to a production SQL Server. You'll need to download each custom report and then re-upload them into the production system.

To download your custom reports from SQL Server Express:

1. Create a temporary folder on your local computer.

    You'll use this folder to store your downloaded custom reports temporarily.

2. Open Delinea Report Services in Internet Explorer.
3. Navigate to the Custom Reports folder.
4. Select a report and select **Download** from the report's action menu.
5. Save the downloaded report in the temporary folder that you already created.

   Repeat this process for each report.
6. Close Internet Explorer.

To upload your custom reports to your production instance of SQL Server:

1. Run the Delinea Report Services Configuration wizard.
2. In the configuration wizard, choose the production SQL Server instance where you want to deploy the reports, then close the wizard.
3. Open Delinea Report Services in Internet Explorer.
4. Navigate to the Custom Reports folder.
5. For each report:
    1. Click **Upload File** and select the custom report that you downloaded from your other instance.
    2. After the report is uploaded, select the report and click **Manage**.
    3. Click the **Data Sources** tab.
    4. Select **A shared data source** and click **Browse**.
    5. In the folder listing, expand the **Centrify Report Services** folder.
    6. Select **ReportDataSource** and click **OK**.
    7. In the Data Sources page, click **Apply**.

You can now open the custom report successfully using data in your production SQL Server instance.

Views to Use in Custom Reports

Database views provide an easier and more secure way to share the reporting data without having to expose the database tables directly. Each view is essentially a database query. Some columns refer to columns in other views, and these relationships are noted.

Understanding the Differences Between Views

There are many views that are very similar to each other but provide different levels of details related to role assignments and so forth. This section briefly covers the differences between views so that you can decide which view to use, based on your needs.

When choosing which view to use, keep in mind that a view that provides less detail results in a faster query response time.

A list of who can log in to which computers	EffectiveAuthorizedUsers_Computer (Including computers in classic and hierarchical zones) EffectiveAuthorizedUsers_Computer_Classic (Only computers in classic zones) EffectiveAuthorizedUsers_Computer_Hierarchical (Only computers in hierarchical zones
A list of who can log in to which computer and what privileges are granted to these users	EffectiveLoginUserPrivileges_Computer EffectiveAuthorizedUserPrivileges_Computer (Same as EffectiveLoginUserPrivileges_Computer, just to consist the naming as the other views)
A list of Active Directory users' effective role assignments	EffectiveRoleAssignment (Both hierarchical & classic zones) EffectiveRoleAssignment_Classic (Classic zones only) EffectiveRoleAssignment_Hierarchical (Hierarchical zones only)
A list of the Active Directory users' effective privileges at the computer level (The Active Directory users list in the view may not have the access right to the computer)	EffectiveRolePrivileges_Computer
A list of the Active Directory users' effective system rights at the computer level (The Active Directory users list in the view might not have the access right to the computer)	EffectiveSysRights
A list of the authorized users' privileges The list indicates if a role or right supports its accessibility to the computer	EffectiveUserPrivileges_Computer
A list of the Active Directory users' privileges at the computer role level	EffectiveUserPrivileges_ComputerRole_Unix (Assuming all computers managed by the Computer Role are UNIX) EffectiveUserPrivileges_ComputerRole_ Windows (Assuming all computers managed by the Computer Role are Windows)
A list of the Active Directory users' privileges at the Zone level	EffectiveUserPrivileges_Zone_Unix (Assuming all computers managed by the Zone are UNIX) EffectiveUserPrivileges_Zone_Windows (Assuming all computers managed by the Zone are UNIX)
Local users	EffectiveAuthorizedLocalUsers_Computer (A local users' version to the EffectiveAuthorizedUsers_Computer) EffectiveAuthorizedLocalUserPrivileges_Computer (A Local users' version to the EffectiveLoginUserPrivileges_Computer) EffectiveLocalUsersRoleAssignment (A Local users' version to the EffectiveRoleAssignment)

ADComputers View

The ADComputers view lists all Active Directory computers for each monitored domain.

ADComputer_-Account-Enabled	1 – Active Directory computer's account is enabled, 0 – account is disabled

ADComputer_AccountEnabled_Desc	The display value for ADComputer_Role (Yes/No)	
ADComputer_-Canonical-Name	Active Directory computer's canonical name	
ADComputer_CnName	The Active Directory computer's common name.	
ADComputer_-Description	The description to the Active Directory computer	
ADComputer_-Dns-Host-Name	Active Directory computer's dnsHostName	
ADComputer_DomainId	The identification number of the computer's domain.	Domains.Id
ADComputer_Domain-Name	The name of the domain that the Active Directory computer belongs to.	
ADComputer_-GUID	The object GUID of the Active Directory computer	
ADComputer_Location	The Active Directory computer's location.	
ADComputer_ManagerGUID	The hosting Active Directory computer's GUID for the user or group.	
ADComputer_ManagerObjectName	The Active Directory computer's manager object name.	
ADComputer_ManagerType	The type of computer manager. 1=user, 2=group.	
ADComputer_ManagerType_Desc	The description of the Active Directory manager type.	
ADComputer_ObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ADComputer_-OS	Active Directory computer's operating system	
ADComputer_-Os-Version	Active Directory computer's operating system version	
ADComputer_OU	The OU of the Active Directory computer. It will be null if the computer is not under an OU	
ADComputer_PwdLastChangedTime	The last changed time for Active Directory computer's password (UTC time). This is an approximation only.	
ADComputer_Role	Whether the computer is running as a domain controller or not 1 - workstation role, 2 - domain controller role	
ADComputer_Role_Desc	The display value for ADComputer_Role (Workstation/Domain Controller)	
ADComputer_-Sam-Account-Name	Active Directory computer's samAccountName	
ADComputer_-Time-Created	The creation time of the Active Directory computer (UTC time)	
ADComputer_TrustedDelegate	Allows services to act on behalf of another user.	

Adcomputers Columns Used in Other Views

ADComputer_-GUID	ADGroupComputerMembers.ADComputer_GUID ComputerRoleMembership.ADComputer_GUID ZoneComputers.ZoneComputer_ADComputerId

ADComputers_Stale View

The ADComputers_Stale view lists all stale Active Directory computers for each domain. Computers are considered as stale if the passwords for them haven't changed for 90 or more days.

ADComputer_AccountEnabled	1 – Active Directory computer's account is enabled, 0 – account is disabled	
ADComputer_AccountEnabled_Desc	The display value for ADComputer_Role (Yes/No)	
ADComputer_-Canonical-Name	Active Directory computer's canonical name	
ADComputer_CnName	The Active Directory computer's common name.	
ADComputer_-Description	The description about the Active Directory computer	
ADComputer_DnsHostName	Active Directory computer's dnsHostName	
ADComputer_DomainId	The ID of the computer's domain	Domains.Id
ADComputer_DomainName	The name of the domain which the Active Directory computer belongs to	
ADComputer_GUID	The object GUID of the Active Directory computer	
ADComputer_ObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ADComputer_OS	Operating system of Active Directory computer	
ADComputer_OsVersion	The operating system version number of the Active Directory computer.	
ADComputer_OU	The OU of the Active Directory computer. It will be null if the computer is not under an OU	
ADComputer_PwdLastChangedTime	The last changed time for Active Directory computer's password (UTC time). This is an approximation only.	
ADComputer_-Role	Whether the computer is running as a domain controller or not 1 - workstation role, 2 - domain controller role	
ADComputer_Role_Desc	The display value for ADComputer_Role (Workstation/Domain Controller)	
ADComputer_SamAccountName	Active Directory computer's samAccountName	
ADComputer_-Time-Created	The creation time of the Active Directory computer (UTC time)	

ADGroupComputerMembers View

The ADGroupComputerMembers lists all computers that are members for each Active Directory group. Nested members are included.

ADComputer_CanonicalName	The canonical name of the Active Directory computer	
ADComputer_CnName	The Active Directory computer's common name.	
ADComputer_DnsHostName	The DNS host name of the Active Directory computer	

ADComputer_GUID	The GUID of the Active Directory computer	ADComputers.ADComputer_GUID
ADComputer_ObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ADComputer_Os	The operating system name of the Active Directory computer	
ADComputer_OsVersion	The OS version of the Active Directory computer	
ADComputer_SamAccountName	The samAccountName of the Active Directory computer	
ADGroup_CanonicalName	The canonical name of the Active Directory group	
ADGroup_GUID	The GUID of the Active Directory group	ADGroups.GUID
ADGroup_Name	The name of the Active Directory group	
ADGroup_ObjectName	The display name for the Active Directory group, formatted as <group samAccountName>@<domain name>.	

ADGroups View

The ADGroups view lists all Active Directory groups for each domain.

ADGroup_ManagerGUID	The hosting Active Directory computer's GUID for the user or group.	
ADGroup_ManagerObjectName	The object name for the user or group who manages this group.	
ADGroup_ManagerType	The type of object that is the manager for this group. 1=user, 2=group.	
ADGroup_ManagerType_Desc	The description of the Active Directory manager type.	
CanonicalName	Active Directory group's canonical name	
Description	Active Directory group's description	
DomainId	The identification for the domain which the Active Directory group belongs to	Domains.Id
Email	Active Directory group's email	
GroupName	Active Directory group's name	
GUID	The object GUID of the Active Directory group.	
IsBuiltIn	1 – is built in group, 0 – is not built in group	
NTLogonName	The NT logon name (samAccountName) of the Active Directory group	
ObjectName	The display name for the Active Directory group, formatted as <group samAccountName>@<domain name>.	
OU	The OU of the Active Directory group. It is null if the group is not under an OU	

TimeCreated	The creation time of the Active Directory group (UTC time)	
Type	The scope of the Active Directory group 1 - domain local, 2 - global, 3 - universal	

ADGroups columns used in other views

ADGroups.GUID	ADGroupComputerMembers.ADGroup_GUID ADGroupUserMembers.ADGroup_GUID EffectiveZoneGroups.ZoneGroup_ADGroup_GUID ZoneGroups.ZoneGroup_ADGroup_GUID EffectiveUserPrivileges_Computer.Trustee_Id EffectiveUserPrivileges_ComputerRole.Trustee_Id EffectiveUserPrivileges_Zone.Trustee_Id

ADGroupSubGroups View

Lists the Active Directory group and the nested groups, including children groups and grand-children groups.

ParentGroup_CanonicalName	The canonical name of the parent group	
ParentGroup_DomainId	The domainID of the parent group	Domains.Id
ParentGroup_DomainName	The domain name of the parent group	
ParentGroup_GroupType	The group type of the parent group 1-Domain local, 2-Global, 3-Universal	
ParentGroup_GroupTypeDesc	The display value for ParentGroup_GroupType (Domain local/Global/Universal)	
ParentGroup_NTLogonName	The NTLogonName of the parent group	
ParentGroup_ObjectName	The object name of the parent group. The general display value for the AD group in precanned report. Format: < AD group samAccountName > @ < domain Name >	
ParentGroup_ParentGroupGUID	The object GUID of the parent group	ADGroups.GUID
ParentGroup_ParentGroupName	The name of the parent group	
SubGroup_CanonicalName	The canonical name of the sub group	
SubGroup_DomainId	The domainIDof the sub group	Domains.Id
SubGroup_DomainName	The domain name of the sub group	
SubGroup_EffectiveSubGroupGUID	The object GUID of the sub group	ADGroups.GUID
SubGroup_GroupName	The group name of the sub group	
SubGroup_GroupType	The group type of the sub group 1-Domain local, 2-Global, 3-Universal	
SubGroup_GroupTypeDesc	The display value for SubGroup_GroupType (Domain local/Global/Universal)	
SubGroup_NTLogonName	The NTLogon name of the sub group Note: There is also a column with a similar name, SubGroup_NTLogoName, that will be deprecated in a future release.	

SubGroup_ObjectName	The object name of the sub group. The general display value for the AD group in precanned report. Format: <AD group samAccountName>@<domain Name>	

ADGroupUserMembers View

The ADGroupUserMembers view lists all user members for each Active Directory group. Nested members are included.

ADGroup_CanonicalName	The canonical name of the Active Directory group	
ADGroup_GUID	The GUID of the Active Directory group	ADGroups.GUID
ADGroup_Name	The name of the Active Directory group	
ADGroup_ObjectName	The display name for the Active Directory group, formatted as <group samAccountName>@<domain name>.	
ADUser_GUID	The GUID of the Active Directory user	ADUsers.ADUser_GUID
ADUser_Name	The name of the Active Directory user	
ADUser_ObjectName	The object name for the Active Directory user.	
ADUser_SamAccountName	The samAccountName of the Active Directory user	
ADUser_UPN	The upn name of the Active Directory user	

ADUsers View

The ADUsers view lists all Active Directory users for each monitored domain.

ADUser_AccountExpiryDate	The expiration date for the Active Directory user account.	
ADUser_AccountLockedUntil	The date and time until which time that the user's account is locked.	
ADUser_AccountLockedUntil_Desc	The description text string for the ADUser_AccountLockedUntil field.	
ADUser_CannotBeDelegated	Cannot be delegated.	
ADUser_CanonicalName	The canonical name of the Active Directory user	
ADUser_City	The city of the Active Directory user	
ADUser_Company	The company of the Active Directory user	
ADUser_Country	The country of the Active Directory user	
ADUser_CreationTime	The creation time of the Active Directory user	
ADUser_Department	The department of the Active Directory user	
ADUser_Description	The description of the Active Directory user	

ADUser_DialInCallbackNumber	The dialin callback number of the Active Directory user	
ADUser_DialInCallbackOptions	The dialin callback options of the Active Directory user	
ADUser_DialInCallerId	The dialin callerIDof the Active Directory user	
ADUser_DialInStaticIp	The dialin static IP address of the Active Directory user	
ADUser_DialInStaticRoutes	The dialin static routes of the Active Directory user	
ADUser_DisplayName	The display name of the Active Directory user	
ADUser_DomainId	TheIDof the Domain	Domains.Id
ADUser_DomainName	The name of the Domain	
ADUser_Email	The email of the Active Directory user	
ADUser_Enabled	If the Active Directory user account is enabled 1 – Enabled, 0 - Disabled	
ADUser_Enabled_Desc	The description string for the aduser_enabled (Yes / No)	
ADUser_FaxNumbers	The fax numbers of the Active Directory user	
ADUser_FirstName	The first name of the Active Directory user	
ADUser_GUID	The GUID of the Active Directory user	
ADUser_HomePhoneNumbers	The home phone numbers of the Active Directory user	
ADUser_Initials	The initials of the Active Directory user	
ADUser_IpPhoneNumbers	The ip phone numbers of the Active Directory user	
ADUser_IsNeverExpire	Specifies if the user account is set to never expire.	
ADUser_IsNeverExpire_Desc	The description text string for the ADUser_IsNeverExpire column.	
ADUser_JobTitle	The job title of the Active Directory user	
ADUser_LastLogonTime	The last logon time of the Active Directory user	
ADUser_LastName	The last name of the Active Directory user	
ADUser_LogonScriptPath	The logon script path of the Active Directory user	
ADUser_ManagerGUID	The hosting Active Directory user's GUID of the user or group	
ADUser_ManagerObjectName	The Active Directory user's manager object name	
ADUser_ManagerType	The Active Directory user's manager type 1 - User, 2-Group	
ADUser_ManagerType_Desc	The Active Directory user's manager type description (User/Group)	
ADUser_MobilePhoneNumbers	The mobile phone numbers of the Active Directory user	

ADUser_Name	The name of the Active Directory user	
ADUser_ObjectName	The display name for the Active Directory user, formatted as <user samAccountName>@<domain name>.	
ADUser_Office	The office of the Active Directory user	
ADUser_PagerPhoneNumbers	The pager phone numbers of the Active Directory user	
ADUSer_PasswordNeverExpire	Password set to never expire.	
ADUser_PhoneNumbers	The phone numbers of the Active Directory user	
ADUser_PoBox	The post office box address of the Active Directory user.	
ADUser_PostalCode	The postal code (zip code) of the Active Directory user.	
ADUser_PreauthenticationNotRequired	Pre-authentication not required.	
ADUser_PrimaryGroupId	The primary group ID of the Active Directory group.	
ADUser_ProfileHomeFolder	The profile home folder of the Active Directory user	
ADUser_ProfilePath	The profile path of the Active Directory user	
ADUser_PwdLastSetTime	The password last set time of the Active Directory user. This is an approximation only.	
ADUser_PwdStoreUsingReversibleEncryption	Password stored using reversible encryption.	
ADUser_RemoteAccessPermissions	The remote access permissions of the Active Directory user	
ADUser_SamAccountName	The samAccountName of the Active Directory user	
ADUser_SmartCardNeededForLogon	Smart card needed for login.	
ADUser_State	The state of the Active Directory user	
ADUSer_Street	The Active Directory user's street address.	
ADUser_TrustedForDelegation	Trusted for delegation.	
ADUser_Upn	The upn name of the Active Directory user	
ADUser_UseDesEncryption	Uses DES Encryption.	
ADUser_WebPages	The web pages of the Active Directory user	

ADUser Columns Used in Other Views

ADUsers.ADUser_GUID	ADGroupUserMembers.ADUser_GUID EffectiveUserPrivileges_Computer.ADUser_GUID EffectiveUserPrivileges_ComputerRole.ADUser_GUID EffectiveUserPrivileges_Zone.ADUser_GUID EffectiveZoneUsers.ZoneUser_ADUserGUID ZoneUsers.ZoneUser_ADUserGUID EffectiveUserPrivileges_Computer.Trustee_Id EffectiveUserPrivileges_ComputerRole.Trustee_Id EffectiveUserPrivileges_Zone.Trustee_Id

ApplicationRight View

The ApplicationRight view lists the detailed attributes for each application right.

Right_Description	The description of the application right	
Right_FullName	The full name of the right <right name>/<zone name>	
Right_GUID	The GUID of the Right	Rights.Right_GUID
Right_Name	The name of the application right	
Right_Priority	The priority of the application right	
Right_RequireAuthentication	If this right requires authentication 1 – Yes, 0 – No	
Right_RequireAuthentication_Desc	If this right requires authentication (Yes/No)	
Right_RunasUser	Run as the specified AD user	
Right_ZoneId	The Id of the Zone that the Right belongs to	Zones.Zone_Id
Right_ZoneName	The name of the Zone that the Right belongs to	

AutoZoneComputers View

The AutoZoneComputers view lists the computers that are joined to the AutoZone.

ZoneComputer_ADComputerCnName	AD computer's cn name	
ZoneComputer_ADComputerId	The GUID of the AD computer	ADComputers_ADComputer_GUID
ZoneComputer_ADComputerName	AD computer's name	
ZoneComputer_ADComputerObjectName	Format: <AD computer CN>.<AD computer domain> Mainly used by precanned-report	
ZoneComputer_AgentVersion	The agent version of the Auto Zone Computer	
ZoneComputer_ComputerType	The IDof the computer type of the Auto Zone Computer. This value is always 2	
ZoneComputer_ComputerType_Desc	The computer type of the Auto Zone Computer. This value is always 'Unix'	
ZoneComputer_Id	The ID of the Auto Zone Computer	
ZoneComputer_IsOrphan	To identifier if this is an orphan Auto Zone Computer 1 – Yes, 0 – No	
ZoneComputer_IsOrphan_Desc	(Yes/No)	

ZoneComputer_Name	The name of the Auto Zone Computer	
ZoneComputer_ZoneId	The ID of the zone. Always be -1	
ZoneComputer_ZoneName	The name of the zone. The value is always 'Auto Zone'	

CommandRight View

This view lists the detailed attributes for each command right.

Right_AddVar	Comma separated list of environment variable name-value pairs to add	
Right_AllowNested	Nested command execution is allowed or not 1 – Yes, 0 – No	
Right_AllowNested_Desc	The description to the Right_AllowNested (Yes/No)	
Right_Authentication	Type of authentication required to run the command	
Right_DeleteVar	Comma separated list of environment variables to delete in addition to the default set	
Right_Description	The description of the command right	
Right_DzdoRunAsGroup	Comma separated list of groups allowed to run this command using dzdo	
Right_DzdoRunAsUser	Comma separated list of users, uids, groups or gids allowed to run this command using dzdo	
Right_DzshRunas	The user this command will run as under dzsh	
Right_FullName	The full name of the command rights. Format <command right name>/<zone name>	
Right_GUID	The GUID of the command right	Rights.Right_GUID
Right_KeepVar	Comma separated list of environment variables to keep in addition to the default set	
Right_MatchPath	The match path of the command right	
Right_Name	The name of the command right	
Right_Pattern	The pattern of the command right	
Right_PatternType	The type of the command right pattern 0 – Global, 1 – Regular expression	
Right_PatternType_Desc	The description of the type of the command right pattern (Global / Regular expression)	
Right_PreserveGroup	Preserve group membership or not	
Right_Priority	The priority of the command right	
Right_UMask	The umask value used to define who can execute the command	
Right_ZoneId	The ID of the zone that the command right is defined	Zones.Zone_Id
Right_ZoneName	The name of the zone that the command right is defined	

ComputerRoleCustomAttribute View

This view lists the computer role custom attributes.

RoleAssignment_GUID	The computer role's object GUID.	ComputerRoles.ComputerRole_GUID
CustomAttribute_Name	The custom attribute's name.	
CustomAttribute_Value	The custom attribute's value.	

ComputerRoleEffectiveMembers View

This view lists the effective members of a computer role.

ComputerRole_GUID	The GUID of the Computer Role	
ComputerRole_ZoneId	The zone ID where the Computer Role is defined	Zones.Zone_Id
ComputerRole_ComputerRoleName	The name of the Computer Role	
ADComputer_GUID	The object GUID of the Active Directory computer	ADComputes.ADComputer_GUID
ADComputer_DomainId	The ID of the computer's domain	Domains.Id
ADComputer_ObjectName	Format: <AD computer CN>.<AD computer domain> This field is mainly used by the default reports.	
ADComputer_CnName	The Active Directory computer's cnName	
ADComputer_DnsHostName	The DNS host name of the Active Directory computer	
ZoneComputer_Id	The ID of the computer	
ZoneComputer_ZoneId	The ID of the zone that the computer is managed by	Zones.Zone_Id
ZoneComputer_Name	The name of the computer	
ZoneComputer_AgentVersion	The agent version of the computer	
ZoneComputer_Platform	The platform of the computer 1 – Windows, 2 – UNIX	
ZoneComputer_Platform_Desc	The description string of the ZoneComputer_Platform (Windows/UNIX)	
ZoneComputer_IsOrphan	If the computer is orphan 1 – Yes, 0 – No	
ZoneComputer_JoinDate	The date when the computer joined zone (UTC time)	

ComputerRoleMembership View

The ComputerRoleMembership view lists all computer members for each Computer Role. The view includes computers that have been added into the zone.

ADComputer_CnName	The Active Directory computer's common name.	
ADComputer_DnsHostName	The dns host name of the Active Directory Computer	
ADComputer_DomainId	The domain ID of the Active Directory computer	Domains.Id
ADComputer_GUID	The GUID of the Active Directory computer	ADComputes.ADComputer_GUID
ADComputer_ObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ComputerRole_ComputerRoleName	The name of the Computer Role	
ComputerRole_GUID	The object GUID of the computer role	
ComputerRole_ZoneId	The ID of the zone where this computer role is defined	Zones.Zone_Id
ZoneComputer_AgentVersion	The agent version of the computer	
ZoneComputer_Id	The ID of the computer	
ZoneComputer_IsOrphan	If the computer is orphaned 1 – Yes, 0 – No	
ZoneComputer_JoinDate	The date when the computer joined zone (UTC time)	
ZoneComputer_Name	The name of the computer	
ZoneComputer_Platform	The computer platform 1 – Windows, 2 – Unix	
ZoneComputer_PlatformDesc	The display value of ZoneComputer_Platform (Windows/Unix)	
ZoneComputer_ZoneId	The ID of the zone where the computer is joined to	Zones.Zone_Id

ComputerRoles View

This view lists the computer role information.

ComputerRole_Description	The description of the Computr Role	
ComputerRole_GroupGUID	The GUID of the AD group which the Computer Role monitoring	ADGroups.GUID
ComputerRole_GroupName	The name of the AD group which the Computer Role monitoring	
ComputerRole_GUID	The GUID of the Computer Role	
ComputerRole_Name	The name of the Computer Role	
ComputerRole_ZoneId	The ID of the zone where the Computer Role is defined	Zones.Zone_Id
ComputerRole_ZoneName	The name of the zone where the Computer Role is defined	

DelegationTasks View

This view lists which user, group, computer, or well-known SID have which delegation tasks.

Target	The target in which the Server Suite task is delegated	
Target_DomainId	The domain ID of the target	Domains.Id
Target_GUID	The GUID of the target	
Zone_Id	The zone ID	Zones.Zone_Id
Scope	The scope in which the Server Suite task is delegated	
Scope_Id	The scope ID: 1 - Zone; 2 - UNIX Computer; 3 - WINDOWS Computer; 4 - Computer Role	
Trustee_Name	The trustee name	
Trustee_Type	The trustee type is one of the following: 1 - User; 2 - UNIX computer; 3 - Windows computer; 4 - computer role.	
Trustee_Type_Desc	The description of the trustee type	
Trustee_DomainId	The domain ID of the trustee	Domains.Id
Task_Id	Task Id	DelegationTaskType.Task_Id
Task_Name	Task name	

DelegationTaskType View

This view lists the Server Suite delegation tasks.

Task_Name	Task name	
Task_Id	Task Id	

Domains View

The Domains view lists all monitored domains.

Dc	The domain controller for the monitored domain
DomainName	The name of the monitored domain
Id	The ID of the monitored domain

Domains Columns Used in Other Views

Domains.Id	ADComputers.ADComputer_DomainID ADComputers_Stale.ADComputer_DomainId ADGroups.DomainId ADUsers.ADUser_DomainID ComputerRoleMembership.ADComputer_DomainId RoleAssignments_ComputerRole.RoleAssignment_ZoneDomainId UserAccounts.ADUser_DomainId ZoneRolePrivileges.ZoneRolePrivileges_RightZoneDomainId Zones.Zone_DomainID Zones_Classic.Zone_DomainID Zones_Hierarchical.Zone_DomainID

EffectiveAuthorizedUserPrivilegesSummary View

This view lists effective privileges rights granted to Active Directory users for both hierarchical and classic zones.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_ID	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id
Right_GUID	The GUID of the right	Rights.Right_GUID

EffectiveAuthorizedUserPrivilegesSummary__Hierarchical View

This view lists effective privileges rights granted to Active Directory users for just hierarchical zones.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_ID	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id
Right_GUID	The GUID of the right	Rights.Right_GUID

EffectiveAuthorizedUserPrivilegesSummary_Classic View

This view lists effective privileges rights granted to Active Directory users for just classic zones.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_ID	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id
Right_GUID	The GUID of the right	Rights.Right_GUID

EffectiveAuthorizedLocalUserPrivileges_Computer View

This view lists the authorized local user's effective rights and privileges for each computer.

ADComputer_CanonicalName	The canonical name of the Active Directory computer	
ADComputer_CnName	The cn name of the Active Directory computer	
ADComputer_DnsHostName	The dns host name of the Active Directory computer	
ADComputer_ObjectName	The object name of the Active Directory computer	

Assigned_Location	The display value of the source assignment location	
Assigned_LocationType	The source assignment location	
Assigned_LocationType_Desc	The type of the source assignment location 1 – Zone 2 – Computer 3 – Computer Role	
EffectiveZone_Id	The auto generated ID of the Zone	Zones.Zone_Id
EffectiveZone_Name	The name of the Zone	
LocalUser_Name	The name of the local user	
LocalUser_ProfileState	The profile state of the local user 1 =Enabled, 2 = Disabled, 3 = Removed from /etc/passwd	
LocalUser_ProfileState_Desc	The display value for LocalUser _ProfileState (Enabled/Disabled/Removed from /etc/passwd)	
Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_Grants_Logon	If this right could support a user to logon to a system 1 – Yes, 0 – No	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The ID of the right platform	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The type ID of the right	RightType.RightTypeId
Right_Type_Desc	The type description of the right	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
Trustee_Name	The trustee name of the role assignment	
Trustee_Type	The trustee type ID of the role assignment	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The type description of the trustee	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id

EffectiveAuthorizedLocalUsers_Computer View

This view lists the effective, authorized local users for each computer.

LocalUser_Name	The name of the local user	ZoneLocalUsers.ZoneLocalUser_Name
ZoneComputer_Id	The ID of the zone computer	ZoneComputers.ZoneComputer_Id
LocalUserProfileState	The state of the local user profile, indicated by a number: Enabled Disabled Removed from /etc/passwd	
LocalUser_ProfileState_Desc	The text description of LocalUserProfileState	

EffectiveAuthorizedUserPrivileges_Computer View

This view lists the users who are authorized to log in and the computers that they can log in to. This EffectiveAuthorizedUserPrivileges_Computer view is the same as EffectiveLoginUserPrivilege_Computer View .

EffectiveAuthorizedUsers_Computer View

This view lists the users who can log in and the computers that they can log in to.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_Id	The computer profile's object GUID.	ZoneComputer.ZoneComputer_ID

EffectiveAuthorizedUsers_Computer_Classic View

This view lists the users who can log in and the classic zone computers that they can log in to.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_Id	The computer profile's object GUID.	ZoneComputer.ZoneComputer_ID

EffectiveAuthorizedUsers_Computer_Hierarchical View

This view lists the users who can log in the hierarchical zone computers that they can log in to.

ADUser_GUID	The object GUID of the Active Directory user that the user profile refers to.	ADUsers.ADUser_GUID
ZoneComputer_Id	The computer profile's object GUID.	ZoneComputer.ZoneComputer_ID

EffectiveAuthorizedZoneLocalUsers View

This view lists the effective user profiles for local users who can log in and the computers that they can log in to.

EffectiveZone_Id	The auto generated ID of the Zone	Zones.Zone_Id
EffectiveZone_Name	The name of the Zone	

EffectiveZone_DomainId	The domain ID of the Zone	
ZoneLocalUser_Id	The auto generated ID of the local user profile	ZoneLocalUsers. ZoneLocalUser_Id
ZoneLocalUser_Name	The name of the local user profile	
ZoneLocalUser_HomeDirectory	The home directory of the local user profile	
ZoneLocalUser_PrimaryGroupId	The primary group ID of the local user profile	
ZoneLocalUser_PrimaryGroupName	The primary group name of the local user profile	
ZoneLocalUser_Shell	The shell of the local user profile	
ZoneLocalUser_Uid	The UID of the local user profile	
ZoneLocalUser_GECOS	The GECOS of the local user profile	
ZoneLocalUser_ProfileState	The profile state of the local user profile 1 means Enabled, 2 means Disabled, 3 means Removed from /etc/passwd	
ZoneLocalUser_ProfileState_Desc	The display value for ZoneLocalUser_ProfileState (Enabled/Disabled/Removed from /etc/passwd)	
ZoneLocalUser_AssignmentLocation_Type	The type code of the location where the zoned local user is assigned	
ZoneLocalUser_AssignmentLocation_Type_Desc	The display text of the type of the location where the zoned local user is assigned	
ZoneLocalUser_AssignmentLocation_GUID	The GUID of the location object where the zoned local user is assigned	
ZoneLocalUser_AssignmentLocation_Name	The name of the location object where the zoned local user is assigned	
ZoneComputer_Id	The object GUID of the computer profile	ZoneComputers. ZoneComputer_Id
ADComputer_ObjectName	The object name of the ad computer	
ADComputer_DnsHostName	The DNS host name of the ad computer	
ADComputer_CnName	The CN name of the ad computer	
ADComputer_Os	The operating system of the Active Directory computer	
ADComputer_DomainId	The domain ID of the Active Directory computer	

EffectiveAuthorizedZoneUsers View

This view lists the authorized Active Directory user's effective user profiles for each computer.

EffectiveZone_Id	The auto-generated ID of the Zone	Zones.Zone_Id

EffectiveZone_Name	The name of the Zone	
EffectiveZone_DomainId	The domain ID of the Zone	
ZoneUser_Id	The auto generated ID of the user profile	ZoneUsers. ZoneUser_Id
ZoneUser_Name	The name of the user profile	
ZoneUser_HomeDirectory	The home directory of the user profile	
ZoneUser_PrimaryGroupId	The primary group ID of the user profile	
ZoneUser_PrimaryGroupName	The primary group name of the user profile	
ZoneUser_Shell	The shell of the user profile	
ZoneUser_Uid	The UID of the user profile	
ZoneUser_GECOS	The GECOS of the user profile	
ZoneUser_IsSecondaryProfile	Whether the user profile is a secondary profile or not: 1 – Yes 0 – No	
ZoneUser_IsSecondaryProfile_Desc	The display value for ZoneUser_IsSecondaryProfile (Yes/No)	
ZoneUser_AssignmentLocation_Type	The type code of the location where the zoned user is assigned	
ZoneUser_AssignmentLocation_Type_Desc	The display text of the type of the location where the zoned user is assigned	
ZoneUser_AssignmentLocation_GUID	The GUID of the location object where the zoned user is assigned	
ZoneUser_AssignmentLocation_Name	The name of the location object where the zoned user is assigned	
ADUser_DomainId	The domain ID of the Active Directory user	
ADUser_GUID	The GUID of the ad user	
ADUser_ObjectName	The object name of the Active Directory user	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id
ADComputer_ObjectName	The object name of the Active Directory computer	
ADComputer_DnsHostName	The DNS host name of the Active Directory computer	
ADComputer_CnName	The CN name of the Active Directory computer	
ADComputer_Os	The operating system of the Active Directory computer	
ADComputer_DomainId	The domain ID of the Active Directory computer	

EffectiveDelegationTasks View

This view lists which Active Directory user has which delegation tasks.

Target	The target in which the Server Suite task is delegated	
Target_DomainId	The domain ID of the target	Domains.Id
Target_GUID	The GUID of the target	
Zone_Id	The zone ID	Zones.Zone_Id
Scope	The scope in which the Server Suite task is delegated	
Scope_Id	The scope ID: 1 - Zone; 2 - UNIX Computer; 3 - WINDOWS Computer; 4 - Computer Role	
Trustee_Name	The trustee name	
Trustee_GUID	The GUID of trustee	
Trustee_DomainId	The domain ID of the trustee	Domains.Id
Task_Id	Task Id	DelegationTaskType.Task_Id
Task_Name	Task name	

EffectiveGroupPrivileges_Computer View

This view lists the consolidated role assignments, logon privileges, system rights privileges for each group and computer. This view only lists the role assignments that are assigned to Active Directory groups, and lists the trustee Active Directory groups and nested groups.

ADComputer_CanonicalName	The canonical name of the Active Directory Computer in where the privileges effective	
ADComputer_CnName	The CN name of the Active Directory Computer in where the privileges effective	
ADComputer_DnsHostName	The DNS host name of the Active Directory Computer in where the privileges effective	
ADComputer_ObjectName	The object name of the Active Directory Computer in where the privileges effective	
ADGroup_CanonicalName	The canonical name of the effective assigned Active Directory group	
ADGroup_GUID	The GUID of the effective assigned Active Directory group	ADGroups.GUID
ADGroup_Name	The name of the effective assigned Active Directory group	
ADGroup_ObjectName	The object name of the effective assigned Active Directory group. The format is ‹samAccountName›@‹domain name›	
ADGroup_SamAccountName	The samAccountName of the effective assigned Active Directory group	
Assigned_Location	The name of the assignment location	
Assigned_LocationType	The type of the assignment location 1 – Zone, 2 – Computer, 3 – Computer Role	

Assigned_LocationTypeDesc	The description fo the type of the assignment location (Zone, Computer, Computer Role)	
Computer_Platform	The platform ID of the Active Directory Computer in where the privileges effective 1 – Windows, 2 – UNIX	
Computer_Platform_Desc	The platform description name of the Active Directory Computer in where the privileges effective (Windows/UNIX)	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_FullName	The full name of the right	
Right_Grants_Logon	If this right could support a user to logon to a system 1 – Yes, 0 – No	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The platform ID of the right 0 – Windows, 1 – UNIX, 2 – Windows/UNIX	
Right_Platform_Desc	The platform description of the right (Windows, UNIX, Windows/UNIX)	
Right_Type	The type ID of the right	RightType.RightTypeId
Right_Type_Desc	The type description of the right	
Role_FullName	The full name of the role <role name>/<zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
Trustee_Id	The GUID of the Trustee	ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type ID of the trustee type	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The type description of the trustee type	
ZoneComputer_Id	The ID of the Zone Computer in where the privileges effective	ZoneComputer.ZoneComputer_Id

EffectiveLocalUserPrivilegesSummary View

This view lists effective privileges rights granted to local UNIX users for both hierarchical and classic zones.

LocalUser_Name	The name of the local user	ZoneLocalUsers.ZoneLocalUser_Name

LocalUser_ProfileState	The state of the local user profile, indicated by number: Enabled Disabled Removed from /etc/passwd	
LocalUser_ProfileState_Desc	The text description of LocalUser_ProfileState.	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputers. ZoneComputer_Id
Right_GUID	The GUID of the right	Rights.Right_GUID

EffectiveLocalUsersRoleAssignment View

This view lists the effective role assignments for local users for each computer.

Assigned_Location	The name of the assigned location	
Assigned_LocationTypeDesc	The assigned location: zone, computer, or computer role	
LocalUser_Name	The name of the local user	ZoneLocalUsers.ZoneLocalUser_Name
LocalUser_ProfileState	The state of the local user profile, indicated by number: Enabled Disabled Removed from /etc/passwd	
LocalUser_ProfileState_Desc	The text description of LocalUser_ProfileState.	
Role_GUID	The GUID for the role.	Roles.Role_Id
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Name	The trustee name	
Trustee_Type	The type of trustee, indicated by number: Active Directory user Active Directory group Local UNIX user Local UNIX group Local Windows user Local Windows group All Active Directory users All local UNIX users All local Windows users local UNIX UID	
Trustee_Type_Desc	The text description of the Trustee_Type	
ZoneComputer_Id	The ID of the zone computer.	ZoneComputers.ZoneComputer_Id

EffectiveLoginUserPrivilege_Computer View

This view lists the users who can log in and the computers that they can log in to. .

ADComputer_CanonicalName	The canonical name of the AD Computer in where the privileges effective	

ADComputer_CnName	The Cn name of the AD Computer in where the privileges effective	
ADComputer_DnsHostName	The dns host name of the AD Computer in where the privileges effective	
ADComputer_ObjectName	The object name of the AD Computer in where the privileges effective	
ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when trustee type = 7	ADUsers.ADUser_GUID
ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The display name for the Active Directory user, formatted as ‹user samAccountName›@‹domain name›.	
ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The upn name of the assigned Active Directory user. It will be null when trustee type = 7	
Assigned_Location	The name of the assignment location	
Assigned_LocationType	The type of the assignment location 1 – Zone, 2 – Computer, 3 – Computer Role	
Assigned_LocationTypeDesc	The description fo the type of the assignment location (Zone, Computer, Computer Role)	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_FullName	The full name of the right	
Right_Grants_Logon	If this right could support a user to logon to a system 1 – Yes, 0 – No	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The platform ID of the right 0 – Windows, 1 – UNIX, 2 – Windows/UNIX	
Right_Platform_Desc	The platform description of the right (Windows, UNIX, Windows/UNIX)	

Right_Type	The type ID of the right	RightType.RightTypeId
Right_Type_Desc	The type description of the right	
Role_FullName	The full name of the role <role name>/<zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
Trustee_ID	The ID of the Trustee	Trustee_Type = 1: ADUsers.ADUser_GUID Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type ID of the trustee type	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The type description of the trustee type	
ZoneComputer_Id	The ID of the Zone Computer in where the privileges effective	ZoneComputer.ZoneComputer_Id

EffectiveRoleAssignment View

This view lists all effective role assignments for each user and for each computer.

ADUser_GUID	The object GUID of the AD user which the user profile referring to.	ADUsers.ADUser_GUID
Assigned_Location	The source assignment location	
Assigned_LocationType	The type of the source assignment location 1 – Zone 2 – Computer 3 – Computer Role	
Assigned_LocationType_Desc	The display value of the source assignment location	
Role_GUID	The object GUID ID of the role	Roles.Role_Id
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The trustee ID of the role assignment	
Trustee_Name	The trustee name of the role assignment	
Trustee_Type	The trustee type ID of the role assignment	TrusteeTypes.TrusteType_Id
Trustee_Type_Desc	The type description of the trustee	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputer.ZoneComputer_Id

EffectiveRoleAssignment_Classic View

This view lists all effective role assignments in classic zones for each user and for each computer.

ADUser_GUID	The object GUID of the AD user which the user profile referring to.	ADUsers.ADUser_GUID
Assigned_Location	The source assignment location	
Assigned_LocationType	The type of the source assignment location 1 – Zone 2 – Computer 3 – Computer Role	
Assigned_LocationType_Desc	The display value of the source assignment location	
Role_GUID	The object GUID ID of the role	Roles.Role_Id
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The trustee ID of the role assignment	
Trustee_Name	The trustee name of the role assignment	
Trustee_Type	The trustee type ID of the role assignment	TrusteeTypes.TrusteType_Id
Trustee_Type_Desc	The type description of the trustee	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputer.ZoneComputer_Id

EffectiveRoleAssignment_Hierarchical View

This view lists all effective role assignments in hierarchical zones for each user and for each computer.

ADUser_GUID	The object GUID of the AD user which the user profile referring to.	ADUsers.ADUser_GUID
Assigned_Location	The source assignment location	
Assigned_LocationType	The type of the source assignment location 1 – Zone 2 – Computer 3 – Computer Role	
Assigned_LocationType_Desc	The display value of the source assignment location	
Role_GUID	The object GUID ID of the role	Roles.Role_Id
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	

RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The trustee ID of the role assignment	
Trustee_Name	The trustee name of the role assignment	
Trustee_Type	The trustee type ID of the role assignment	TrusteeTypes.TrusteType_Id
Trustee_Type_Desc	The type description of the trustee	
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputer.ZoneComputer_Id

EffectiveRolePrivileges_Computer View

This view lists the consolidated role assignments, logon privileges, system rights privileges for each computer. This view does not expand the trustee to individual Active Directory users.

ADComputer_CanonicalName	The canonical name of the AD Computer in where the privileges effective	
ADComputer_CnName	The Cn name of the AD Computer in where the privileges effective	
ADComputer_DnsHostName	The dns host name of the AD Computer in where the privileges effective	
ADComputer_ObjectName	The object name of the AD Computer in where the privileges effective	
Assigned_Location	The name of the assignment location	
Assigned_LocationType	The type of the assignment location 1 – Zone, 2 – Computer, 3 – Computer Role	
Assigned_LocationTypeDesc	The description fo the type of the assignment location (Zone, Computer, Computer Role)	
Computer_Platform	The platform ID of the AD Computer in where the privileges effective 1 – Windows, 2 – UNIX	
Computer_Platform_Desc	The platform description name of the AD Computer in where the privileges effective (Windows/UNIX)	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_Description	The description of the right.	
Right_FullName	The full name of the right	
Right_Grants_Logon	If this right could support a user to logon to a system 1 – Yes, 0 – No	

Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The platform ID of the right 0 – Windows, 1 – UNIX, 2 – Windows/UNIX	
Right_Platform_Desc	The platform description of the right (Windows, UNIX, Windows/UNIX)	
Right_Type	The type ID of the right	RightType.RightTypeId
Right_Type_Desc	The type description of the right	
Role_FullName	The full name of the role <role name>/<zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
Trustee_GUID	The GUID of the Trustee	Trustee_Type = 1: ADUsers.ADUser_GUID Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type ID of the trustee type	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The type description of the trustee type	
ZoneComputer_Id	The ID of the Zone Computer in where the privileges effective	ZoneComputer.ZoneComputer_Id

EffectiveSysRights View

This view lists the effective system rights in hierarchical zones for each user and for each computer.

ADUser_GUID	The object GUID of the AD user which the user profile referring to.	ADUsers.ADUser_GUID
ZoneComputer_Id	The object GUID ID of the computer profile	ZoneComputer.ZoneComputer_Id
-Audit-Level	The role's audit level (It will be null for classic zone's role) 0 – audit not required, 1 – audit if possible, 2 – audit required	
AuditLevel_Desc	The display value of Role_AuditLevel (It will be null for classic zone's role) (Audit not Required/Audit if Possible/Audit required)	
Always-Permit-Logon	(It will be null for classic zone's role) 1 – always permit, 0 – not always permit	
AlwaysPermitLogon_Desc	The display value of -Always-Permit-Logon (It will be null for classic zone's role) (Always permit/Not always permit)	
AllowPasswordLogon	Allow Password Logon 0 – No, 1 – Yes, Null – N/A	

AllowPasswordLogon_Desc	The display value of AllowPasswordLogon (No, Yes, N/A)	
AllowPsRemoteAccess	Allow PowerShell remote access 0 - No, 1- Yes, Null - N/A	
AllowPsRemoteAccess_Desc	The display value of AllowPsRemoteAccess (No, Yes, N/A)	
AllowNonPasswordLogon	Allow Non Password Logon 0 – No, 1 – Yes, Null – N/A	
AllowNonPasswordLogon_Desc	The display value of AllowNonPasswordLogon (No, Yes, N/A)	
AllowConsoleLogon	Allow Console Logon 0 – No, 1 – Yes, Null – N/A	
AllowConsoleLogon_Desc	The display value of AllowConsoleLogon (No, Yes, N/A)	
AllowRemoteLogon	Allow Remote Logon 0 – No, 1 – Yes, Null – N/A	
AllowRemoteLogon_Desc	The display value of AllowRemoteLogon (No, Yes, N/A)	
HasVisibleRight	Has Visible Right 0 – No, 1 – Yes, Null – N/A	
HasVisibleRight_Desc	The display value of HasVisibleRight (No, Yes, N/A)	
IgnoreDisabled	If this user has 'ignore disabled' right on this computer 0 – No, 1 – Yes, Null – N/A	
IgnoreDisabled_Desc	The display value of IgnoreDisabled (No, Yes, N/A)	

EffectiveUserPrivileges_Computer View

The EffectiveUserPrivileges_Computer view lists consolidated role assignments, logon privileges, and system rights' privileges for each user and computer.

ADComputer_CanonicalName	The canonical name of the computer	
ADComputer_CnName	The Active Directory computer's common name.	
ADComputer_DnsHostName	The DNS host name of the computer	
ADComputer_ObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when trustee type = 7	ADUsers.ADUser_GUID
ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The display name for the Active Directory user, formatted as <user samAccountName>@<domain name>.	

ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The upn name of the assigned Active Directory user. It will be null when trustee type = 7	
Assigned_Location	The name of the source assignment location. It might be the zone name, computer dns host name or Computer Role name, depends on the location type	
Assigned_LocationType	The type of the source assignment location 1 – Zone 2 – Computer 3 – Computer Role	
Assigned_LocationTypeDesc	The display value of the source assignment location Zone Computer Computer Role	
Effective_AllowConsoleLogon	If this user has 'console logon' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_AllowLogon	If this user can logon this computer	
Effective_AllowNonPasswordLogon	If this user has 'non password logon' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_AllowNonRestrictedShell	If this user has 'non restricted Shell' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_AllowPasswordLogon	If this user has 'password logon' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_AllowPsRemoteAccess	If this user has the 'PowerShell Remote Access' right on this computer 0 - No; 1 - Yes; Null - N/A	
Effective_AllowRemoteLogon	If this user has 'remote logon' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_AuditLevel	The human readable text of the effective audit level for this user on this computer 0 – Audit not required, 1 –Audit if possible, 2 – Audit required	
Effective_CloudAuthorizationRequired	If this user has 'Cloud authorization required' right on this computer 0 – No, 1 – Yes, Null – N/A	
Effective_HasRescueRight	If this role grants 'rescue' right to this user on this computer 0 – No, 1 – Yes	
Effective_HasVisibleRight	Specifies if the user is visible on this computer	
Effective_IgnoreDisabled	If this user has 'ignore disabled' right on this computer 0 – No, 1 – Yes, Null – N/A	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id Zones_Hierarchical.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	

Grants_AuditLevel	If this role grants the effective Audit level 0 – Audit not required, 1 – Audit if possible, 2 – Audit required Given the Effective AuditLevel is 0 If this roles's AuditLevel equals to the Effective Audit Level, then this column is 1 – Yes, Otherwise, 0 -- No	
Grants_CloudAuthorizationRequired	If this role grants 'Cloud authorization required' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_ConsoleLogon	If this role grants 'console logon' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_HasVisibleRight	Specifies if the role grants the visible right to this user on this computer.	
Grants_IgnoreDisabled	If this role grants 'ignore disabled' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_Logon	If this role grants logon	
Grants_NonPasswordLogon	If this role grants 'non password logon' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_NonRestrictedShell	If this role grants 'non restricted Shell' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_PasswordLogon	If this role grants 'password logon' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_PsRemoteAcccess	If this role grants the 'PowerShell Remote Access' right to this user on this computer 0 - No; 1 - Yes; Null - N/A	
Grants_RemoteLogon	If this role grants 'remote logon' right to this user on this computer 0 – No, 1 – Yes, Null – N/A	
Grants_RescueRight	If this user has 'rescue' right on this computer 0 – No, 1 – Yes	
Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	Whether the right applies to windows, unix or both.	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The ID of the right type	RightType.RightTypeId
Right_Type_Desc	The display value of the right type (see RightTypes view)	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	

RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
Trustee_Id	The GUID of the trustee	Trustee_Type = 1: ADUsers.ADUser_GUID Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type of the trustee 1 – Active Directory users 2 – Active Directory groups 7 – All Active Directory users	
Trustee_Type_Desc	The display value of the trustee Active Directory users Active Directory groups All Active Directory users	
ZoneComputer_Id	The zone computer ID	ZoneComputer.ZoneComputer_Id

EffectiveUserPrivileges_ComputerRole_UNIX View

The EffectiveUserPrivileges_ComputerRole_UNIX view lists effective computer role level role assignments for each user. This view assumes that all computers within the computer role are UNIX computers. The assigned Active Directory users must have at least one completed profile in the zone where the computer role is defined. Assignee "All Active Directory users" will be expanded to Active Directory users.

ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when trustee type = 7	ADUsers.ADUser_GUID
ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The general display value for the Active Directory use in the default report. The format is <Active Directory samAccountName>@<domain name>.	
ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The upn name of the assigned Active Directory user. It will be null when trustee type = 7	
Assigned_Location	The name of the source assignment location. For this view, it will be always the Computer Role name	
Assigned_LocationType	The type of the source assignment location 3 – Computer Role	
Assigned_LocationTypeDesc	The display value of the source assignment location Computer Role	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id Zones_Hierarchical.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	

Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The ID of the right platform	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The ID of the right type	RightType.RightTypeId
Right_Type_Desc	Whether this right is for Unix, Windows or both	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The GUID of the trustee	If Trustee_Type = 1: ADUsers.ADUser_GUID If Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type of the trustee 1 – Active Directory users 2 – Active Directory groups 7 – All Active Directory users	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The display value of the trustee Active Directory users Active Directory groups All Active Directory users	

Note: Assigned_LocationType and Assigned_LocationTypeDesc might be removed in subsequent release.

EffectiveUserPrivileges_ComputerRole_Windows View

The EffectiveUserPrivileges_ComputerRole_Windows view lists effective computer role level role assignments for each user. This view assumes that all computers within the computer role are Windows computers. Assignee "All Active Directory users" are NOT expanded to Active Directory users.

ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when trustee type = 7	ADUsers.ADUser_GUID

ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The general display value for the Active Directory use in the default report. The format is <Active Directory samAccountName>@<domain name>.	
ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The upn name of the assigned Active Directory user. It will be null when trustee type = 7	
Assigned_Location	The name of the source assignment location. For this view, it will be always the Computer Role name	
Assigned_LocationType	The type of the source assignment location 3 – Computer Role	
Assigned_LocationTypeDesc	The display value of the source assignment location Computer Role	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id Zones_Hierarchical.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	The ID of the right platform	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The ID of the right type	RightType.RightTypeId
Right_Type_Desc	Whether this right is for Unix, Windows or both	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The GUID of the trustee	If Trustee_Type = 1: ADUsers.ADUser_GUID If Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type of the trustee 1 – Active Directory users 2 – Active Directory groups 7 – All Active Directory users	TrusteeTypes.TrusteeType_Id

Trustee_Type_Desc	The display value of the trustee Active Directory users Active Directory groups All Active Directory users	

EffectiveUserPrivileges_Zone_UNIX View

The EffectiveUserPrivileges_Zone view lists effective zone level role assignments for each user. This view assumes that all computers in the zone are UNIX computers. The assigned Active Directory users must have at least one completed profile in the zone. Assignee "All Active Directory users" is expanded to Active Directory users.

ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when trustee type = 7	ADUsers.ADUser_GUID
ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The display value for the Active Directory in the default report. The format is <Active Directory samAccountName>@<domain name>.	
ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The upn name of the assigned Active Directory user. It will be null when trustee type = 7	
Assigned_Location	The name of the the source assignment location. For this view, it will be always the same as the EffectiveZone_Name	
Assigned_LocationType	The type of the source assignment location 1 – Zone	
Assigned_LocationTypeDesc	The display value of the source assignment location Zone	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	Whether this right is for Unix, Windows or both	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The ID of the right type	RightType.RightTypeId

Right_Type_Desc	The display value of the right type	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The GUID of the trustee	if Trustee_Type = 1: ADUsers.ADUser_GUID If Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type of the trustee 1 – Active Directory users 2 – Active Directory groups 7 – All Active Directory users	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The display value of the trustee: Active Directory users Active Directory groups All Active Directory users	

Note: Assigned_LocationType and Assigned_LocationTypeDesc may be removed in a subsequent release.

EffectiveUserPrivileges_Zone_Windows View

This view lists the effective role assignments for each user, assuming that all computers within the zone are Windows computers. Assignee "All Active Directory users" is NOT expanded to Active Directory users.

ADUser_CanonicalName	The canonical name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_FullName	The full name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_GUID	The GUID of the assigned Active Directory user. It will be null when the trustee type = 7	ADUsers.ADUser_GUID
ADUser_Name	The name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_ObjectName	The display value for the Active Directory in the default report. The format is <Active Directory samAccountName>@<domain name>.	
ADUser_SamAccountName	The samAccount name of the assigned Active Directory user. It will be null when trustee type = 7	
ADUser_Upn	The UPN name of the assigned Active Directory user. It will be null when trustee type = 7	

Assigned_Location	The name of the the source assignment location. For this view, it will be always the same as the EffectiveZone_Name	
Assigned_LocationType	The type of the source assignment location 1 – Zone	
Assigned_LocationTypeDesc	The display value of the source assignment location Zone	
EffectiveZone_Id	The ID of the effective zone for the privilege assignment	Zones.Zone_Id
EffectiveZone_Name	The name of the effective zone for the privilege assignment	
Right_FullName	The full name of the right. Format in <Right name> / <Right's zone name>	
Right_GUID	The GUID of the right	Rights.Right_GUID
Right_Name	The name of the right	
Right_Platform	Whether this right is for Unix, Windows or both	
Right_Platform_Desc	The display value of the right platform	
Right_Type	The ID of the right type	RightType.RightTypeId
Right_Type_Desc	The display value of the right type	
Role_FullName	The full name of the role. Format in <Role name> / <Role's zone name>	
Role_GUID	The GUID of the role	Roles.Role_Id
Role_Name	The name of the role	
RoleAssignment_GUID	The object GUID of the role assignment	RoleAssignments.RoleAssignment_GUID
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Trustee_Id	The GUID of the trustee	if Trustee_Type = 1: ADUsers.ADUser_GUID If Trustee_Type = 2: ADGroups.ADGroup_GUID
Trustee_Name	The name of the trustee	
Trustee_Type	The type of the trustee 1 – Active Directory users 2 – Active Directory groups 7 – All Active Directory users	TrusteeTypes.TrusteeType_Id
Trustee_Type_Desc	The display value of the trustee: Active Directory users Active Directory groups All Active Directory users	

EffectiveZoneGroups View

The EffectiveZoneGroups view lists effective group profiles for each computer and zone.

ZoneGroup_ADGroupGUID	The object GUID of the Active Directory group which the group profile referring to.	ADGroups.GUID
ZoneGroup_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneGroup_AssignmentLocation_Name	The name of the assignment location	
ZoneGroup_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneGroup_AssignmentLocation_TypeDesc	(zone/Computer)	
ZoneGroup_Gid	The GID of the group profile	
ZoneGroup_Id	The auto generated ID of the group profile	ZoneGroups.ZoneGroup_Id
ZoneGroup_Name	The UNIX name of the group	
ZoneGroup_ZoneComputerId	The ID of the computer where the group profile is effective	ZoneComputers.ZoneComputer_Id
ZoneGroup_ZoneId	The ID of the zone where the group profile is defined	Zones.Zone_Id

EffectiveZoneLocalGroupMembers View

This view lists the effective local group members for each computer and zone.

ZoneLocalGroup_ZoneId	The ID of the zone where the local group profile under	Zones.Zone_Id
ZoneLocalGroup_ZoneComputerId	The ID of the computer profile where the local group profile effective in	ZoneComputers.ZoneComputer_Id
ZoneLocalGroup_Name	The UNIX name of the local group	
ZoneLocalGroup_MemberName	The name of the local group's member	
ZoneLocalGroup_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneLocalGroup_AssignmentLocation_TypeDesc	(zone/Computer)	
ZoneLocalGroup_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneLocalGroup_AssignmentLocation_Name	The name of the assignment location	

EffectiveZoneLocalGroups View

This view lists the effective local group profiles for each computer and zone.

ZoneLocalGroup_Id	The auto generated ID of the local group profile	ZoneLocalGroups.ZoneLocalGroup_Id
ZoneLocalGroup_ZoneId	The ID of the zone where the local group profile under	Zones.Zone_Id

ZoneLocalGroup_ZoneComputerId	The ID of the computer profile where the local group profile effective in	ZoneComputers.ZoneComputer_Id
ZoneLocalGroup_Name	The UNIX name of the group	
ZoneLocalGroup_Gid	The GID of the local group profile	
ZoneLocalGroup_ProfileState	The profile state of the local group profile 1 = Enabled, 3 = Removed from /etc/group	
ZoneLocalGroup_ProfileState_Desc	The display value for ZoneLocalGroup_ProfileState (Enabled/Removed from /etc/group)	
ZoneLocalGroup_IsCompleteProfile	To indicate if this profile was a complete profile 1 – Yes, 0 - No	
ZoneLocalGroup_IsCompleteProfile_Desc	The description to the ZoneLocalGroup_IsCompleteProfile (Yes/No)	
ZoneLocalGroup_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneLocalGroup_AssignmentLocation_TypeDesc	(zone/Computer)	
ZoneLocalGroup_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneLocalGroup_AssignmentLocation_Name	The name of the assignment location	

EffectiveZoneLocalUsers View

This view lists the effective local user profiles for each computer and zone.

ZoneLocalUser_Id	The auto generated ID of the local user profile	ZoneLocalUsers.ZoneLocalUser_Id
ZoneLocalUser_ZoneId	The ID of the zone where the local user profile under	Zones.Zone_Id
ZoneLocalUser_ComputerProfileId	The name of the zone where the local user profile under	ZoneComputers.ZoneComputer_Id
ZoneLocalUser_HomeDirectory	The local user profile's home directory	
ZoneLocalUser_Name	The local user profile's unix name	
ZoneLocalUser_PrimaryGroupId	The local user profile's primary group id	
ZoneLocalUser_PrimaryGroupName	The local user profile's primary group name	
ZoneLocalUser_GECOS	The local user profile's GECOS	
ZoneLocalUser_Shell	The local user profile's shell	
ZoneLocalUser_Uid	The local user profile's UID	
ZoneLocalUser_ProfileState	The profile state of the local user 1= Enabled, 2 = Disabled, 3 = Removed from /etc/passwd	

ZoneLocalUser_ProfileState_Desc	The display value for ZoneLocalUser_ProfileState (Enabled/Disabled/Removed from /etc/passwd)	
ZoneLocalUser_IsCompleteProfile	To indicate if this profile was a complete profile 1 – Yes, 0 - No	
ZoneLocalUser_IsCompleteProfile_Desc	The description to the ZoneLocalUser_ IsCompleteProfile (Yes/No)	
ZoneLocalUser_AssignmentLocation_Type	The type code of the location where the zoned user is assigned	
ZoneLocalUser_AssignmentLocation_TypeDesc	The display text of the type of the location where the zoned local user is assigned	
ZoneLocalUser_AssignmentLocation_GUID	The GUID of the location object where the zoned local user is assigned	
ZoneLocalUser_AssignmentLocation_Name	The name of the location object where the zoned local user is assigned	

EffectiveZoneLocalWinGroupMembers View

This view lists the effective local Windows group members for each computer and zone.

ZoneLocalGroup_ZoneId	The ID for the zone that contains the local Windows group profile	Zones.Zone_Id
ZoneLocalGroup_ZoneComputerId	The ID of the computer profile where the local Windows group profile is effective	ZoneComputers.ZoneComputer_Id
ZoneLocalGroup_Name	The name of the local Windows group	
ZoneLocalGroup_MemberName	The name of the local Windows group's member	
ZoneLocalGroup_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneLocalGroup_AssignmentLocation_TypeDesc	The display value for ZoneLocalGroup_AssignmentLocation_Type (zone/Computer)	
ZoneLocalGroup_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneLocalGroup_AssignmentLocation_Name	The name of the assignment location	

EffectiveZoneLocalWinGroups Views

This view lists the effective local Windows group profiles for each computer and zone.

ZoneLocalGroup_Id	The auto-generated ID of the local Windows group profile	ZoneLocalGroups.ZoneLocalGroup_Id
ZoneLocalGroup_ZoneId	The ID of the zone that contains the local Windows group profile	Zones.Zone_Id

ZoneLocalGroup_ZoneComputerId	The ID of the computer profile where the local Windows group profile is effective	ZoneComputers.ZoneComputer_Id
ZoneLocalGroup_Name	The name of the local Windows group	
ZoneLocalGroup_Description	The description of the local Windows group	
ZoneLocalGroup_ProfileState	The profile state of the local Windows group profile 1 = Enabled, 3 = Removed	
ZoneLocalGroup_ProfileState_Desc	The display value for ZoneLocalGroup_ProfileState (Enabled/Removed)	
ZoneLocalGroup_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneLocalGroup_AssignmentLocation_TypeDesc	The display value for ZoneLocalGroup_AssignmentLocation_Type(zone/Computer)	
ZoneLocalGroup_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneLocalGroup_AssignmentLocation_Name	The name of the assignment location	

EffectiveZoneLocalWinUsers View

This view lists the effective local Windows user profiles for each computer and zone.

ZoneLocalUser_Id	The auto-generated ID of the local windows user profile	ZoneLocalUsers.ZoneLocalGroup_Id
ZoneLocalUser_ZoneId	The ID of the zone that contains the local Windows user profile	Zones.Zone_Id
ZoneLocalUser_ComputerProfileId	The ID of the computer profile where the local Windows user profile is effective	ZoneComputers.ZoneComputer_Id
ZoneLocalUser_Name	The name of the local Windows user	
ZoneLocalUser_FullName	The full name of the local Windows user	
ZoneLocalUser_PasswordOption	The password option of the local Windows user	
ZoneLocalUser_Description	The description of the local Windows user	
ZoneLocalUser_ProfileState	The profile state of the local Windows user profile 1 = Enabled, 2 = Disabled, 3 = Removed	
ZoneLocalUser_ProfileState_Desc	The display value for ZoneLocalUser_ProfileState (Enabled/Disabled/Removed)	
ZoneLocalUser_AssignmentLocation_Type	The type code of the assignment location type 1 – Zone, 2 – Computer	
ZoneLocalUser_AssignmentLocation_TypeDesc	The display value for ZoneLocalUser_AssignmentLocation_Type (zone/Computer)	

ZoneLocalUser_AssignmentLocation_GUID	The object GUID of the assignment location	
ZoneLocalUser_AssignmentLocation_Name	The name of the assignment location	

EffectiveZoneUsers View

The EffectiveZoneUsers view lists effective user profiles for each computer and zone,

ZoneUser_ADUserGUID	The object GUID of the Active Directory user which the user profile referring to.	ADUsers.ADUser_GUID
ZoneUser_AssignmentLocation_GUID	The GUID of the location object where the zoned user is assigned	
ZoneUser_AssignmentLocation_Name	The name of the location object where the zoned user is assigned	
ZoneUser_AssignmentLocation_Type	The type code of the location where the zoned user is assigned	
ZoneUser_AssignmentLocation_TypeDesc	The display text of the type of the location where the zoned user is assigned	
ZoneUser_ComputerProfileId	The name of the zone computer where the user profile is effective	ZoneComputers.ZoneComputer_Id
ZoneUser_GECOS	The user profile's GECOS	
ZoneUser_HomeDirectory	The user profile's home directory	
ZoneUser_Id	The auto generated ID of the user profile	ZoneUsers.ZoneUser_Id
ZoneUser_IsCompleteProfile	To indicate if this profile was a complete profile 1 – Yes, 0 - No	
ZoneUser_IsCompleteProfile_Desc	The description string for ZoneUser_ IsCompleteProfile (Yes/No)	
ZoneUser_IsEnabled	To indicate if this profile was enabled. Only available to classic zone's profile. For hierarchical zone profile, it will always be null 1 – Yes, 0 - No	
ZoneUser_IsEnabled_Desc	The description string for ZoneUser_ IsEnabled (Yes/No)	
ZoneUser_IsOrphan	1 – It is an orphan user profile. 0 – It is not an orphan profile 1 – Yes, 0 - No	
ZoneUser_IsOrphan_Desc	The description to the ZoneUser_ IsOrphan (Yes/No)	
ZoneUser_IsSecondaryProfile	To indicate if this profile was a secondary profile 1 – Yes, 0 - No	
ZoneUser_IsSecondaryProfile_Desc	The description string for ZoneUser_IsSecondaryProfile (Yes/No)	
ZoneUser_Name	The user profile's unix name	
ZoneUser_PrimaryGroupId	The user profile's primary group id	
ZoneUser_PrimaryGroupName	The user profile's primary group name	
ZoneUser_Shell	The user profile's shell	

ZoneUser_Uid	The user profile's uid	
ZoneUser_ZoneId	The ID of the zone where the user profile under	Zones.Zone_Id

Rights View

The Rights view lists all rights and system rights defined for each zone.

Grants_Logon	Specifies whether the right allows a user to log on to a computer.	
Right_-Description	The description of the right	
Right_-Full-Name	The full name of the right. The format of the full name is: Right_-Name/Right_ZoneName	
Right_-GUID	The object GUID of the right	
Right_-Type	The ID of the right type 1 – Network Access right 2 – Desktop right 3 – Application right 4 – PAM Access right 5 – SSH right 6 – Command right 7 – Restricted Environment 101 – Allow password logon 102 – Allow non password logon 103 – Ignore disabled 104 – Allow non restricted shell 105 – Allow console logon 106 – Allow remote logon 107 – Always permit logon 108 – Audit level – Not required 109 – Audit level – If possible 110 – Audit level – Required 111 – Cloud Authorization Required	RightType.RightTypeId
Right_Type_Desc	The display value of the right type: Network Access right Desktop right Application right PAM Access right SSH right Command right Restricted Environment Allow password logon Allow non password logon Ignore disabled Allow non restricted shell Allow console logon Allow remote logon Always permit logon Audit level – Not required Audit level – If possible Audit level – Required Cloud Authorization Required	
Right_ZoneId	The zone ID of the right. It will be null for system rights	Zones.Zone_Id
Right_-ZoneName	The zone name of the right. It will be null for system rights	

Rights columns used in other views

Rights.Right_GUID	EffectiveUserPrivileges_Computer.Right_GUID EffectiveUserPrivileges_ComputerRole.Right_GUID EffectiveUserPrivileges_Zone.Right_GUID

RightType View

The RightType view provides the type of rights that are defined in the zone and what operating system platform the type applies to.

Grants_Logon	Specifies if the right can support a user to log on to a system. 0 – No 1 – Yes
RightPlatformId	The platform ID of the right type 0 – Unix 1 – Windows 2 – Unix/Windows
RightTypeDesc	The display value of the right type
RightTypeId	The ID of the right type

RightType columns used in other views

RightType.RightTypeId	EffectiveUserPrivileges_Computer.Right_Type EffectiveUserPrivileges_ComputerRole.Right_Type EffectiveUserPrivileges_Zone.Right_Type Rights.Right_Type ZoneRolePrivileges.ZoneRolePrivileges_RightType

RoleAssignmentCustomAttribute View

This view lists the role assignment's custom attributes.

Role_Id	The role's object GUID.	Roles.Role_Id
CustomAttribute_Name	The custom attribute's name.	
CustomAttribute_Value	The custom attribute's value.	

RoleAssignments View

This view lists the role assignments, based on zones, computer roles, or computers.

RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_GUID	The object GUID of the role assignment	
RoleAssignment_StartTime	The start date and time for the role assignment.	
RoleAssignment_ZoneId	The ID of the zone where the role assignment belongs to	Zones.Zone_Id
RoleAssignment_ZoneName	The name of the zone where the role assignment belongs to	
RoleAssignment_ZoneDomainId	The ID of the domain where the role assignment belongs to	Domains.Id
Assigned_Location	The name of the location where the role assignment is created	
Assigned_LocationType	The type of the location where the role assignment was created 1 - Zone , 2 - Computer, 3 - Computer Role	
Assigned_LocationType_Desc	The type description of the location where the role assignment was created 1 - Zone , 2 - Computer, 3 - Computer Role	
RoleAssignment_TrusteeName	The name of the trustee	
RoleAssignment_TrusteeType	The type of the trustee 1 - AD user 2 - AD group 3 - Local UNIX user 4 - Local UNIX group 5 - Local Windows user 6 - Local Windows group 7 - All AD users 8 - All UNIX user 9 - All Windows users	
RoleAssignment_TrusteeType_Desc	The type description of the trustee	
RoleAssignment_RoleGUID	The object GUID of the assigned role	Roles.Role_Id

RoleAssignment_RoleName	The name of the assigned role	
RoleAssignment_RoleFullName	The full name of the assigned role	
RoleAssignment_Description	The description of the role assignment	

RoleAssignments_Computer View

This view lists the role assignments defined for computers.

RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_GUID	The object GUID of the role assignment	
RoleAssignment_ZoneComputerId	The ID of the zone computer where the role assignment is defined	ZoneComputers.ZoneComputer_Id
RoleAssignment_ADComputer_ObjectName	The name of the zone computer where the role assignment is defined	
RoleAssignment_ADComputer_CnName	The Active Directory computer's CN name of the zone computer where the role assignment is defined	
RoleAssignment_ADComputer_CanonicalName	The Active Directory computer's canonical name of the zone computer where the role assignment is defined	
RoleAssignment_ADComputer_DnsHostName	The Active Directory computer's Dns host name name of the zone computer where the role assignment is defined	
RoleAssignment_StartTime	The start date and time for the role assignment.	
Computer_Platform	The platform of the zone computer where the role assignment is defined 1 - Windows 2 - UNIX	.
Computer_Platform_Desc	The platform description of the zone computer where the role assignment is defined 1 - Windows 2 - UNIX	
RoleAssignment_ZoneId	The ID of the zone where the role assignment belongs	Zones.Zone_Id
RoleAssignment_ZoneName	The name of the zone where the role assignment belongs	
RoleAssignment_ZoneDomainId	The ID of the domain where the role assignment belongs	Domains.Id
RoleAssignment_TrusteeName	The name of the trustee	
RoleAssignment_TrusteeType	The type of the trustee 1 - AD user 2 - AD group 3 - Local UNIX user 4 - Local UNIX group 5 - Local Windows user 6 - Local Windows group 7 - All AD users 8 - All UNIX user 9 - All Windows users	
RoleAssignment_TrusteeType_Desc	The type description of the trustee .	
RoleAssignment_RoleGUID	The object GUID of the assigned role	Roles.Role_Id
RoleAssignment_RoleName	The name of the assigned role	

RoleAssignment_RoleFullName	The full name of the assigned role	
RoleAssignment_Description	The role assignment description	

RoleAssignments_ComputerRole View

The RoleAssignments_Computer Role view lists the role assignments for each computer role.

RoleAssignment_ComputerRoleDescription	The description of the Compute Role	
RoleAssignment_ComputerRoleGUID	The GUID of the Computer Role	
RoleAssignment_ComputerRoleName	The name of the Computer Role	
RoleAssignment_Description	The description of the role assignment	
RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_GUID	The object GUID of the role assignment	
RoleAssignment_RoleFullName	The effective end time of the role assignment	
RoleAssignment_RoleGUID	The GUID of the assigned role	Roles.Role_Id
RoleAssignment_RoleName	The object GUID of the role that is being assigned	
RoleAssignment_StartTime	The start date and time for the role assignment.	
RoleAssignment_TrusteeName	The trustee name of the role assignment	
RoleAssignment_TrusteeType	The trustee type code of the role assignment 1 – Active Directory user 2 – Active Directory group 3 – Local UNIX user 4 – Local UNIX group 5 – Local Windows user 6 – Local Windows group 7 – All Active Directory users 8 – All UNIX user 9 – All Windows users	
RoleAssignment_TrusteeType_Desc	The display value of the trustee type: Active Directory user Active Directory group Local UNIX user Local UNIX group Local Windows user Local Windows group All Active Directory users All UNIX user All Windows users	
RoleAssignment_ZoneDomainId	The zone's domain ID of the role assignment	Domains.Id
RoleAssignment_ZoneId	The zone ID of the role assignment	Zones.Zone_Id

RoleAssignments_Zone View

This view lists the role assignments defined in zones.

RoleAssignment_EndTime	The end date and time for the role assignment.	
RoleAssignment_GUID	The object GUID of the role assignment	

RoleAssignment_StartTime	The start date and time for the role assignment.	
RoleAssignment_ZoneId	The ID of the zone where the role assignment belongs	Zones.Zone_Id
RoleAssignment_ZoneName	The name of the zone where the role assignment belongs	
RoleAssignment_ZoneDomainId	The id of the domain where the role assignment belongs	Domains.Id
RoleAssignment_TrusteeName	The name of the trustee	
RoleAssignment_TrusteeType	The type of the trustee 1 - AD user 2 - AD group 3 - Local UNIX user 4 - Local UNIX group 5 - Local Windows user 6 - Local Windows group 7 - All AD users 8 - All UNIX user 9 - All Windows users	
RoleAssignment_TrusteeType_Desc	The type description of the trustee	
RoleAssignment_RoleGUID	The object GUID of the assigned role	Roles.Role_Id
RoleAssignment_RoleName	The name of the assigned role	
RoleAssignment_RoleFullName	The full name of the assigned role	
RoleAssignment_Description	The description of the role assignment	

RoleCustomAttribute View

This view lists the computer role's custom attributes.

RoleAssignment_GUID	The role assignment's object GUID.	RoleAssignments.RoleAssignment_GUID
CustomAttribute_Name	The custom attribute's name.	
CustomAttribute_Value	The custom attribute's value.	

RoleRights View

This view lists the rights for each role.

Role_GUID	The object GUID ID of the role	Roles.Role_Id
Role_-Name	The name of the role	
Role_-Full-Name	The full name of the role. The format of the full name is: <Role_Name>/<Role_ZoneName>	
Right_GUID	The object GUID of the right	Rights.Right_Id
Right_-Name	The zone name of the right. It will be null for system rights.	
Right_-Full-Name	The full name of the right. The format of the full name is: Right_-Name/Right_ZoneName	
Right_ZoneId	The zone ID of the right. This column is null for system rights.	Zones.Zone_Id

Right_-Type	The ID of the right type 1 – Network Access right 2 – Desktop right 3 – Application right 4 – PAM Access right 5 – SSH right 6 – Command right 7 – Restricted Environment 101 – Allow password logon 102 – Allow non password logon 103 – Ignore disabled 104 – Allow non restricted shell 105 – Allow console logon 106 – Allow remote logon 107 – Always permit logon 108 – Audit level – Not required 109 – Audit level – If possible 110 – Audit level – Required 111 – Cloud Authorization Required	RightType.RightTypeId
Right_Type_Desc	The display value of the right type: Network Access right Desktop right Application right PAM Access right SSH right Command right Restricted Environment Allow password logon Allow non password logon Ignore disabled Allow non restricted shell Allow console logon Allow remote logon Always permit logon Audit level – Not required Audit level – If possible Audit level – Required Cloud Authorization Required	
Right_-Description	The description of the right	
Right_Platform	The platform ID of the right 0 – Windows, 1 – UNIX, 2 – Windows/UNIX	
Right_Platform_Desc	The platform description of the right (Windows, UNIX, Windows/UNIX)	
Right_Grants_Logon	If this right could support a user to logon to a system 1 – Yes, 0 – No	

Roles View

The Roles view lists all roles for each zone. This view is a combined view of the Roles_Classic and Roles_Hierarchical views.

Role_-Always-Permit-Logon	(It will be null for classic zone's role) 1 – always permit, 0 – not always permit	
Role_AlwaysPermitLogon_Desc	The display value of _-Always-Permit-Logon (It will be null for classic zone's role) (Always permit/Not always permit)	
Role_-Audit-Level	The role's audit level (It will be null for classic zone's role) 0 – audit not required, 1 – audit if possible, 2 – audit required	
Role_AuditLevel_Desc	The display value of Role_AuditLevel (It will be null for classic zone's role) (Audit not Required/Audit if Possible/Audit required)	
Role_-Description	The description of the role	
Role_-Full-Name	The full name of the role. The format of the full name is: <Role_Name>/<Role_ZoneName>	
Role_-Id	The object GUID of the role	
Role_-Name	The name of the role	
Role_-Zone-Id	The ID of the zone where the role is defined	Zones.Zone_Id
Role_ZoneName	The name of the zone where the role is defined	

Roles Columns Used In Other Views

Roles.Right_GUID	ZoneRolePrivileges.ZoneRolePrivileges_RightGUID

Roles.Role_Id	EffectiveUserPrivileges_Computer.Role_GUID EffectiveUserPrivileges_ComputerRole.Role_GUID EffectiveUserPrivileges_Zone.Role_GUID RoleAssignments_ComputerRole.RoleAssignment_RoleGUID ZoneRolePrivileges.ZoneRolePrivileges_RoleGUID

Roles_Classic View

The Roles_Classic view lists all roles for each classic zone.

Role_-Always-Permit-Logon	(It will be null for classic zone's role) It is NULL in this view as Audit Level is not applicable in classic zone	
Role_AlwaysPermitLogon_Desc	The display value of Role_-Always-Permit-Logon (It will be null for classic zone's role) It is NULL in this view as Audit Level is not applicable in classic zone	
Role_-Audit-Level	The role's audit level (It will be null for classic zone's role) It is NULL in this view as Audit Level is not applicable in classic zone	
Role_AuditLevel_Desc	The display value of Role_AuditLevel (It will be null for classic zone's role) It is NULL in this view as Audit Level is not applicable in classic zone	
Role_-Description	The description of the role	
Role_-Full-Name	The full name of the role. The format of the full name is: <Role_Name>/<Role_ZoneName>	
Role_-Id	The object GUID of the role	
Role_-Name	The name of the role	
Role_-Zone-Id	The ID of the zone where the role is defined	Zones.Zone_Id
Role_ZoneName	The name of the zone where the role is defined	

Roles_Hierarchical View

The Roles_Hierarchical view lists all roles for each hierarchical zone.

Role_-Always-Permit-Logon	1 – always permit, 0 – not always permit	
Role_AlwaysPermitLogon_Desc	The display value of Role_-Always-Permit-Logon (Always permit/Not always permit)	
Role_-Audit-Level	The role's audit level 0 – audit not required, 1 – audit if possible, 2 – audit required	
Role_AuditLevel_Desc	The display value of Role_AuditLevel (Audit not Required/Audit if Possible/Audit required)	
Role_-Description	The description of the role	
Role_-Full-Name	The full name of the role. The format of the full name is: <Role_Name>/<Role_ZoneName>	
Role_-ID	The object ID of the role	
Role_-Name	The name of the role	

Role_-Zone-Id	The ID of the zone where the role is defined	Zones.Zone_Id
Role_ZoneName	The name of the zone where the role is defined	

TrusteeTypes View

This view lists the role assignment trustee types.

TrusteeType_Id	The type ID of the trustee	
TrusteeType_Desc	The type description of the trustee	

Zone_Classic View

The Zones_Classic view lists all Classic zones.

Zone_AvailableShells	Zone's Available shells	
Zone_CanonicalName	The canonical name of the Zone	
Zone_DefaultGroup	Zone's default group	
Zone_DefaultHomeDirectory	Zone's default home directory	
Zone_DefaultPrimaryGroupId	The default primary group	
Zone_DefaultPrimaryGroupName	The name of the default primary group	
Zone_DefaultShell	Zone's default shell	
Zone_DomainId	The name of the domain which the Active Directory user belongs to	Domains.Id
Zone_DomainName	The ID of the domain which the Active Directory user belongs to	
Zone_Id	The auto generated ID of the Zone	
Zone_IsHierarchical	If the zone was a Hierarchical zone or not 1 – Is Hierarchical Zone, 0 – Classic Zone	
Zone_IsHierarchical_Desc	The display value for Zone_IsHierarchical (Yes/No)	
Zone_IsSFU	If the zone was a SFU zone or not 1 – SFU Zone, 0 – Non SFU Zone	
Zone_IsSFU_Desc	(Yes/No)	
Zone_Name	The name of the Zone	
Zone_NextGID	Zone's next gid	
Zone_NextUID	Zone's next uid	
Zone_NISDomain	Zone's NIS domain	

Zone_ReservedGID	Zone's reserved gid	
Zone_ReservedUID	Zone's reserved uid	
Zone_SFUDomain	Zone's SFU domain	

Zone_Hierarchical View

The Zones_Hierarchical view lists all Hierarchical zones.

Zone_AvailableShells	Zone's Available shells	
Zone_CanonicalName	The canonical name of the Zone	
Zone_DefaultGroup	Zone's default group	
Zone_DefaultHomeDirectory	Zone's default home directory	
Zone_DefaultPrimaryGroupId	The default primary group	
Zone_DefaultPrimaryGroupName	The name of the default primary group	
Zone_DefaultShell	Zone's default shell	
Zone_DomainId	The name of the domain which the Active Directory user belongs to	Domains.Id
Zone_DomainName	The ID of the domain which the Active Directory user belongs to	
Zone_Id	The auto generated ID of the Zone	
Zone_IsHierarchical	If the zone was a Hierarchical zone or not 1 – Is Hierarchical Zone, 0 – Classic Zone	
Zone_IsHierarchical_Desc	The display value for Zone_IsHierarchical 1 – Yes, 0 - No	
Zone_IsSFU	If the zone was a SFU zone or not 1 – SFU Zone, 0 – Non SFU Zone	
Zone_IsSFU_Desc	1 – Yes, 0 - No	
Zone_Name	The name of the Zone	
Zone_NextGID	Zone's next gid	
Zone_NextUID	Zone's next uid	
Zone_NISDomain	Zone's NIS domain	
Zone_ReservedGID	Zone's reserved gid	
Zone_ReservedUID	Zone's reserved uid	
Zone_SFUDomain	Zone's SFU domain	
Zone_TrustedCloudInstanceUrl	Trusted cloud instance URL	

Zones_Hierarchical Columns Used in Other Views

Zones_Hierarchical.Zone_Id	EffectiveUserPrivileges_Computer.EffectiveZone_Id EffectiveUserPrivileges_Computer.ZoneUser_Id EffectiveUserPrivileges_ComputerRole.EffectiveZone_Id

ZoneComputers View

The ZoneComputers view lists computer profiles for each zone.

ZoneComputer_ADComputerCnName	The Active Directory computer's common name.	
ZoneComputer_ADComputerDnsHostName	The Active Directory computer's DNS hostname	
ZoneComputer_ADComputerDomainId	The domain ID of the Active Directory computer which is managed by the zone	
ZoneComputer_ADComputerId	The object GUID of the Active Directory computer which is managed by the zone	ADComputers.ADComputer_GUID
ZoneComputer_ADComputerName	The name of the Active Directory computer which is managed by the zone	
ZoneComputer_ADComputerObjectName	The object name of the computer, in the format of <computer CN>.<computer domain>.	
ZoneComputer_AgentVersion	The agent version of the managed computer	
ZoneComputer_ComputerType	The type of the managed computer 1 – Windows, 2 – Unix	
ZoneComputer_ComputerType_Desc	The display value of the ZoneComputer_ComputerType (Windows/Unix)	
ZoneComputer_Id	The object GUID of the computer profile	
ZoneComputer_IsHierarchical	1 – It is managed by a hierarchical zone, 0 – It is managed by a classic zone	
ZoneComputer_IsHierarchical_Desc	The display value of the ZoneComputer_IsHierarchical (Yes/No)	
ZoneComputer_IsOrphan	1 – It is an orphan profile, 0 – It is not an orphan profile	
ZoneComputer_IsOrphan_Desc	The display value of the ZoneComputer_IsOrphan (Yes/No)	
ZoneComputer_IsZoned	If the computer joined zone 1 – Joined zone, 0 – Only has machine overrides	
ZoneComputer_JoinDate	The date when the managed computer joined zone (UTC time)	

ZoneComputer_LicenseType	Specifies the type of computer license. 1 - Server, 2-Workstation, 3-UNIX, 4-Express	
ZoneComputer_LicenseType_Desc	The description of the license type.	
ZoneComputer_Name	The name of the managed computer	
ZoneComputer_PreferredSite	The preferred site of the computer.	
ZoneComputer_PreferredSubnetSite	The preferred subnet site of the computer.	
ZoneComputer_ZoneDomainId	The domain ID of the zone by which the computer is managed	
ZoneComputer_ZoneId	The ID of the zone by which the computer managed	Zones.Zone_Id
ZoneComputer_ZoneName	The name of the zone by which the computer managed	

ZoneComputer Columns Used in Other Views

ZoneComputer.ZoneComputer_Id	EffectiveUserPrivileges_Computer.ZoneComputer_Id EffectiveZoneGroups.ZoneGroup_ZoneComputerId EffectiveZoneUsers.ZoneUser_ComputerProfileId

ZoneGroups View

The ZoneGroups view lists group profiles for each zone.

ZoneGroup_-ADGroup-GUID	The object GUID of the Active Directory group which the group profile referring to.	ADGroups.GUID
ZoneGroup_ADGroupName	The name of the Active Directory group which the user profile referring to.	
ZoneGroup_-Gid	The group profile's gid	
ZoneGroup_-Id	The auto generated ID of the group profile	
ZoneGroup_IsOrphan	If the zone group referencing to a valid Active Directory group 1 – It is an orphan user profile. 0 – It is not an orphan profile	
ZoneGroup_IsOrphan_Desc	The display value for ZoneGroup_IsOrphan 1 – Yes, 0 – No	
ZoneGroup_-Name	The group profile's name	
ZoneGroup_-Zone-Id	The ID of the zone where the group profile is defined	Zones.Zone_Id
ZoneGroup_ZoneName	The name of the zone where the group profile is defined	

ZoneGroup Columns Used in Other Views

ZoneGroups.ZoneGroup_Id	EffectiveUserPrivileges_Computer.ZoneComputer_Id EffectiveZoneGroups.ZoneGroup_ZoneComputerId EffectiveZoneUsers.ZoneUser_ComputerProfileId	

ZoneHierarchy View

ParentZone_Id	The ID of the parent zone.	Zones.Zone_Id
ParentZone_Name	The name of the parent zone.	
ParentZone_DomainID	The domain ID of the parent zone.	Domains.Id
ChildZone_Id	The ID of the child zone.	Zones.Zone_Id
ChildZone_Name	The name of the child zone.	
ChildZone_DomainId	The domain ID of the child zone.	Domains.Id

ZoneLocalGroupMembers View

This view lists the local group members for each zone.

ZoneLocalGroup_-Id	The auto generated ID of the local group profile	
ZoneLocalGroup_-Zone-Id	The ID of the zone where the local group profile is	Zones.Zone_Id
ZoneLocalGroup_ZoneName	The name of the zone where the local group profile is	
ZoneLocalGroup_-Name	The local group profile's name	
ZoneLocalGroup_MemberName	The name of the local group's member	

ZoneLocalGroups View

This view lists the local group profiles for each zone.

ZoneLocalGroup_-Id	The auto generated ID of the local group profile	
ZoneLocalGroup_-Zone-Id	The ID of the zone where the local group profile is	Zones.Zone_Id
ZoneLocalGroup_ZoneName	The name of the zone where the local group profile is	
ZoneLocalGroup_-Gid	The local group profile's GID	
ZoneLocalGroup_-Name	The local group profile's name	
ZoneLocalGroup_ProfileState	The profile state of the local group profile 1 = Enabled, 3 = Removed from /etc/group	
ZoneLocalGroup_ProfileState_Desc	The display value for ZoneLocalGroup_ProfileState (Enabled/Removed from /etc/group)	

ZoneLocalUsers View

This view lists the local user profiles for each zone.

ZoneLocalUser_Id	The auto generated ID of the local user profile	
ZoneLocalUser_ZoneId	The ID of the zone where the local user profile is	Zones.Zone_Id
ZoneLocalUser_ZoneName	The name of the zone where the local user profile is	
ZoneLocalUser_Name	The local user profile's UNIX name	
ZoneLocalUser_HomeDirectory	The local user profile's home directory	
ZoneLocalUser_PrimaryGroupID	The local user profile's primary group ID	
ZoneLocalUser_PrimaryGroupName	The local user profile's primary group name	
ZoneLocalUser_IsHierarchical	If the zone user was defined in a hierarchical zone or not 1 – It is defined in a hierarchical zone. 0 – Is is defined in a classic zone	
ZoneLocalUser_IsHierarchical_Desc	The display value for ZoneLocalUser_IsHierarchical (Yes/No)	
ZoneLocalUser_Shell	The shell of the zone user	
ZoneLocalUser_GECOS	The GECOS of the zone user	
ZoneLocalUser_Uid	The zone user's uid	
ZoneLocalUser_ProfileFlag	The profile state of the local user 1 means Enabled, 2 means Disabled, 3 means Removed from /etc/passwd	
ZoneLocalUser_ProfileFlag_Desc	The display value for ZoneLocalUser_ProfileState (Enabled/Disabled/Removed from /etc/passwd)	

ZoneLocalWinGroupMembers View

This view lists the local Windows group members for each zone.

ZoneLocalGroup_Id	The auto-generated ID of the local Windows group profile	
ZoneLocalGroup_ZoneId	The ID of the zone that contains the local Windows group profile	Zones.Zone_Id
ZoneLocalGroup_ZoneName	The name of the zone that contains the local Windows group profile	
ZoneLocalGroup_Name	The local Windows group profile's name	
ZoneLocalGroup_MemberName	The name of the local Windows group's member	

ZoneLocalWinGroups View

This view lists the local Windows group profiles for each zone.

ZoneLocalGroup_Id	The auto-generated ID of the local Windows group profile	
ZoneLocalGroup_ZoneId	The ID of the zone that contains the local Windows group profile	Zones.Zone_Id
ZoneLocalGroup_ZoneName	The name of the zone that contains the local Windows group profile	
ZoneLocalGroup_Name	The local Windows group profile's name	
ZoneLocalGroup_ProfileState	The profile state of the local Windows group profile 1 = Enabled, 3 = Removed	
ZoneLocalGroup_ProfileState_Desc	The display value for ZoneLocalGroup_ ProfileState (Enabled/Removed)	

ZoneLocalWinUsers View

This view lists the local Windows user profiles for each zone.

ZoneLocalUser_Id	The auto generated ID of the local Windows user profile	
ZoneLocalUser_ZoneId	The ID of the zone where the local Windows user profile is	Zones.Zone_Id
ZoneLocalUser_ZoneName	The name of the zone where the local Windows user profile is	
ZoneLocalUser_Name	The local Windows user profile's name	
ZoneLocalUser_IsHierarchical	If the zone user was defined in a hierarchical zone or not 1 – It is defined in a hierarchical zone. 0 – It is defined in a classic zone	
ZoneLocalUser_IsHierarchical_Desc	The display value for ZoneLocalUser_IsHierarchical (Yes/No)	
ZoneLocalUser_FullName	The full name of the local Windows user	
ZoneLocalUser_Description	The description of the local Windows user	
ZoneLocalGroup_ProfileFlag	The profile state of the local Windows user profile 1 = Enabled, 2 = Disabled, 3 = Removed	
ZoneLocalGroup_ProfileFlag_Desc	The display value for ZoneLocalUser_ ProfileState (Enabled/Disabled/Removed)	

ZoneRolePrivileges View

The ZoneRolePrivileges view lists the roles that are defined for each zone and the rights that are granted by each of these roles.

ZoneRolePrivileges_RightFullName	The full name of the right	
ZoneRolePrivileges_RightGUID	The GUID of the right	Roles.Right_GUID
ZoneRolePrivileges_RightName	The name of the right	
ZoneRolePrivileges_RightPlatform	Whether the right is for Unix, Windows or both	

ZoneRolePrivileges_RightPlatform_Desc	The display value of the right platform	
ZoneRolePrivileges_RightType	The type ID of the right	RightType.RightTypeId
ZoneRolePrivileges_RightType_Desc	The display value of the right's type	
ZoneRolePrivileges_RightZoneDomainId	The domain ID of the zone of the right	Domains.Id
ZoneRolePrivileges_RightZoneId	The zone ID of the right	Zones.Zone_Id
ZoneRolePrivileges_RightZoneIsHierarchical	If the zone of the right is hierarchical 1 – Yes, 0 – No	
ZoneRolePrivileges_RightZoneIsHierarchical_Desc	The display value of the ZoneRolePrivileges_RightZoneIsHierarchical (Yes/No)	
ZoneRolePrivileges_RightZoneName	The zone name of the right	
ZoneRolePrivileges_RoleFullName	The full name of the role	
ZoneRolePrivileges_RoleGUID	The GUID of the role	Roles.Role_Id
ZoneRolePrivileges_RoleName	The name of the role	
ZoneRolePrivileges_RoleZoneDomainId	The domain ID of the zone of the domain	
ZoneRolePrivileges_RoleZoneId	The zone ID of the role	Zones.Zone_Id
ZoneRolePrivileges_RoleZoneIsHierarchical	If the zone of the role is hierarchical 1 – Yes, 0 – No	
ZoneRolePrivileges_RoleZoneIsHierarchical_Desc	The display value of the ZoneRolePrivileges_RoleZoneIsHierarchical (Yes/No)	
ZoneRolePrivileges_RoleZoneName	The zone name of the role	

Zones View

The Zones view lists all the zones in the domain. This view is a combination of both Zones_Classic and Zones_Hierarchical.

Zone_AvailableShells	Zone's Available shells	
Zone_CanonicalName	The canonical name of the Zone	
Zone_DefaultGIDType	The ID of the default GID type 1—Use the auto-incremented GID 2—Use the generated GID from the SID 3—Use the Apple GID scheme	
Zone_DefaultGIDType_Desc	The description of the default GID type (Use auto-incremented GID, Generated GID from SID, or Use Apple GID scheme)	
Zone_DefaultGroup	Zone's default group	
Zone_DefaultHomeDirectory	Zone's default home directory	
Zone_DefaultPrimaryGroupId	The default primary group	

Zone_DefaultPrimaryGroupName	The name of the default primary group	
Zone_DefaultShell	Zone's default shell	
Zone_DefaultUIDType	The ID of the default UID type (applies to hierarchical zones only) 1—Use auto-incremented UID 2—Generated UID from SID 3—Use Apple UID scheme	
Zone_DefaultUIDType_Desc	The description of the default type. For hierarchical zones, this is one of the following: Use auto-incremented UID, Generated UID from SID, or Use Apple UID scheme. For classic zones: Use auto-incremented UID.	
Zone_DefaultUserName	The description of the zone scheme ID, such as Standard, RFC 2307, or SFU.	
Zone_DomainId	The name of the domain which the Active Directory user belongs to	Domains.Id
Zone_DomainName	The ID of the domain which the Active Directory user belongs to	
Zone_Id	The auto generated ID of the Zone	
Zone_IsHierarchical	If the zone was a Hierarchical zone or not 1 – Is Hierarchical Zone, 0 – Classic Zone	
Zone_IsHierarchical_Desc	If the zone was a Hierarchical zone or not (Yes/No)	
Zone_IsSFU	If the zone was a SFU zone or not 1 – SFU Zone, 0 – Non SFU Zone	
Zone_IsSFU_Desc	If the zone was a SFU zone or not (Yes/No)	
Zone_Name	The name of the Zone	
Zone_NextGID	Zone's next gid	
Zone_NextUID	Zone's next uid	
Zone_NISDomain	Zone's NIS domain	
Zone_ReservedGID	Zone's reserved gid	
Zone_ReservedUID	Zone's reserved uid	
Zone_Schema	The ID of the zone scheme: 1—Standard 2—RFC 2307 3—SFU	
Zone_SFUDomain	Zone's SFU domain	
Zone_Type	The zone type (hierarchical or classic)	
Zone_TrustedCloudInstanceUrl	Trusted cloud instance URL	

Zone view columns used in other views

Roles_Classic.Role_ZoneId ComputerRoleMembership.ComputerRole_ZoneId ComputerRoleMembership.ZoneComputer_ZoneId EffectiveUserPrivileges_Computer.EffectiveZone_Id EffectiveUserPrivileges_ComputerRole.EffectiveZone_Id

Zone.Zone_Id	EffectiveUserPrivileges_Zone.EffectiveZone_Id EffectiveZoneGroups.ZoneGroup_ZoneId EffectiveZoneUsers.ZoneUSer_ZoneId Rights.Right_Id RoleAssignments_ComputerRole.RoleAssignment_ZoneId Roles.Role_ZoneId Roles_Hierarchical.Role_ZoneId ZoneComputers.ZoneComputer_ZoneId ZoneGroups.ZoneGroup_ZoneId ZoneRolePrivileges.ZoneRolePrivileges_RoleZoneId ZoneRolePrivileges.ZoneRolePrivileges_RightZoneId ZoneUsers.ZoneUser_ZoneId

ZoneUsers View

The ZoneUsers view lists the user profiles for each zones.

ZoneUser_ADUserGUID	The object GUID of the Active Directory user which the user profile referring to.	ADUsers.ADUser_GUID
ZoneUser_ADUserName	The name of the Active Directory user which the user profile referring to.	
ZoneUser_GECOS	The GECOS of the zone user	
ZoneUser_HomeDirectory	The user profile's home directory	
ZoneUser_Id	The auto generated ID of the user profile	
ZoneUser_IsHierarchical	If the zone user was defined in a hierarchical zone or not 1 – It is defined in a hierarchical zone. 0 – Is is defined in a classic zone	
ZoneUser_IsHierarchical_Desc	The display value for ZoneUser_IsHierarchical (Yes/No)	
ZoneUser_IsOrphan	If the zone user referencing to a valid Active Directory user 1 – It is an orphan user profile. 0 – It is not an orphan profile	
ZoneUser_IsOrphan_Desc	The display value for ZoneUser_IsOrphan (Yes/No)	
ZoneUser_IsSFU	If the zone user was defined in a SFU zone or not 1 – It is defined in a SFU zone. 0 – Is is not defined in a SFU zone	
ZoneUser_IsSFU_Desc	The display value for ZoneUser_IsSFU (Yes/No)	
ZoneUser_Name	The user profile's unix name	
ZoneUser_PrimaryGroupID	The user profile's primary group id	
ZoneUser_PrimaryGroupName	The user profile's primary group name	
ZoneUser_Shell	The shell of the zone user	
ZoneUser_Uid	The zone user's uid	
ZoneUser_UserEnabled	If the zone user is enabled (For classic zone user only, it will be null for Hierarchical zone user) 1 – enabled, 0 – disabled, NULL – not applicable	
ZoneUser_UserEnabled_Desc	(Yes/No)	
ZoneUser_ZoneId	The ID of the zone where the user profile under	Zones.Zone_Id
ZoneUser_ZoneName	The name of the zone where the user profile under	

Delinea

ZoneUser Columns Used In Other Views

ZoneUsers.ZoneUser_Id	EffectiveUserPrivileges_Computer.ZoneUser_Id EffectiveUserPrivileges_ComputerRole.ZoneUser_Id EffectiveUserPrivileges_Zone.ZoneUser_Id EffectiveZoneUsers.ZoneUser_Id

Configuring Report Services for Large Active Directory Environments

Configuration issues can significantly affect the performance of synchronizing Active Directory information and report queries and generation. This section describes additional considerations for deploying Delinea Report Services successfully in a large Active Directory environment.

Memory Recommendations and Requirements for Large Active Directory Environments

Domain Controller Memory

Symptoms

The domain controller runs slower or stops responding.

You can use the Performance monitor tool to evaluate if the system is operating within adequate capacity thresholds.

For details, see: http://social.technet.microsoft.com/wiki/contents/articles/14355.capacity-planning-for-active-directory-domain-services.aspx#Monitoring_For_Compliance_With_Capacity_Planning_Goals

Resolution

Ensure the system has a sufficient amount of RAM. The minimum amount of RAM should be the sum of:

- Active Directory database size (such as the size of the C:\Windows\NTDS\ folder)
- Total SYSVOL size (such as the size of the C:\Windows\SYSVOL folder)
- Operating system recommended amount of RAM
- Vendor recommendations for the agents (antivirus, monitoring, backup, and so on)
- Additional amount of RAM to accommodate growth over the lifetime of the server.

For details, see: http://social.technet.microsoft.com/wiki/contents/articles/14355.capacity-planning-for-active-directory-domain-services.aspx

Windows Memory Requirements

Here are the memory requirements for different versions of Windows:

Windows 2008, 2008 R2	512 MB minimum 2 GB or more is recommended
Windows 2012, 2012 R2	512 MB minimum
Windows 7, 8, 8.1, 10	2 GB minimum for 64-bit systems

References

http://windows.microsoft.com/en-us/windows7/products/system-requirements

http://windows.microsoft.com/en-US/windows-8/system-requirements

http://www.microsoft.com/en-us/windows/windows-10-specifications

https://technet.microsoft.com/en-us/windowsserver/bb414778.aspx

https://technet.microsoft.com/en-us/library/dn303418.aspx

Sql Server Memory

Symptoms

- Delinea Report Services fails to rebuild or refresh a snapshot because of insufficient system memory or an out of memory error.

- You cannot open reports in SSRS because of insufficient system memory or an out of memory error.

Resolution

Ensure that your SQL Server deployment has sufficient memory. Different versions of SQL Server have different memory requirements. For details, please see:

https://msdn.microsoft.com/en-us/library/ms143506.aspx

In addition to Microsoft's recommended memory requirement for SQL Server, an additional amount of memory is required for SQL Server in order to rebuild/refresh snapshot data and render the report successfully.

For more information, see Configuration Recommendations for Large Active Directory Environments.

Configuration Recommendations for Large Active Directory Environments

The major factor of evaluating the configuration requirements for SQL Server is the total number of effective users who can access the computers that are joined to zone in the Active Directory environment. You can estimate the total number of effective users by multiplying the number of computers joined to the zone by the average number of users who can access the computer.

Below lists the recommended configurations for SQL Server for some sample Active Directory environments.

Active Directory environment Sample #1:

Average number of users who can access the computer	500
Total number of effective users	500 * 1000= 500,000
90% of user profiles and role assignments are explicitly defined at the zone level	

Active Directory environment Sample #1 configuration recommendations :

SQL Server memory	8 GB
SQL Server disk space	30 GB

Active Directory environment Sample #2:

Average number of users who can access the computer	3,000
Total number of effective users	3,000 * 5,000 = 15,000,000
90% of user profiles and role assignments are explicitly defined at the zone level	

Active Directory environment Sample #2 configuration recommendations :

SQL Server memory	64 GB
SQL Server disk space	80 GB

Setting the Maximum Server Memory for SQL Server

To prevent Microsoft SQL Server from consuming too much memory, you can use the following formula to determine the recommended maximum server memory:

- Reserve 4GB from the first 16GB of RAM and then 1GB from each additional 8GB of RAM for the operating system and other applications.
- Configure the remaining memory as the maximum server memory allocated for the Microsoft SQL Server buffer pool.

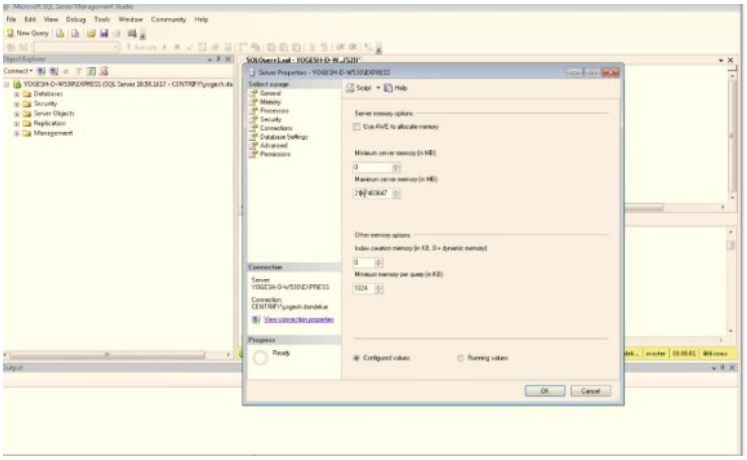
For example, if the computer hosting the Microsoft SQL Server instance has 32GB of total physical memory, you would reserve 4GB (from first 16 GB) + 1GB (from next 8 GB) + 1 GB (from next 8 GB) for the operating system, then set the Maximum server memory for Microsoft SQL Server to 26GB (32GB – 4GB – 1GB – 1GB = 26).

Reference:

https://msdn.microsoft.com/en-us/library/ms178067(v=sql.105).aspx

To set the maximum server memory for SQL Server:

1. Open the SQL Server Management Studio, enter the SQL Server properties:
2. Set the maximum server memory (in MB).



Using Report Filters to Limit the Output Data of a Report

Symptoms

In large Active Directory environments, the following reports can take too long to render because they generate a huge volume of output:

- Authorization Report
- Classic Zone – User Privileged Command Rights Report
- Classic Zone – User Role Assignment Report
- Hierarchical Zone - Computer Role Effective Assignments Report (UNIX)
- Hierarchical Zone - Computer Role Effective Assignments Report (Windows)
- Hierarchical Zone - Effective Audit Level Report
- Hierarchical Zone - Effective Rights Report
- Hierarchical Zone - Effective Role Report
- Hierarchical Zone - Users Report
- Hierarchical Zone - Zone Effective Assignments Report (UNIX)
- Hierarchical Zone - Zone Effective Assignments Report (Windows)
- All PCI reports
- All SOX reports

Resolution

You can use report filters to limit the report to only list data for specific zone types and zones in a specific domain. This can reduce the amount of data output from the report and the report will take less time to render.

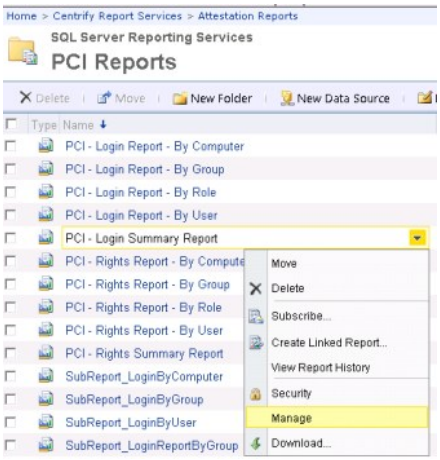
If you are opening the PCI and SOX reports, you can use the Zone Type filter to limit the reports to only list data for Classic zones or Hierarchical zones.

For all reports, you can use the Zone Domain filter to limit the reports to only list data for zones in a specific domain. By default, the Zone Domain filter of all the reports is set to the first zone domain.

By default, reports are set to run automatically when you open the report. If you prefer to set the reports to not run automatically upon opening, do the following. You must have manage report permission in order to configure the report.

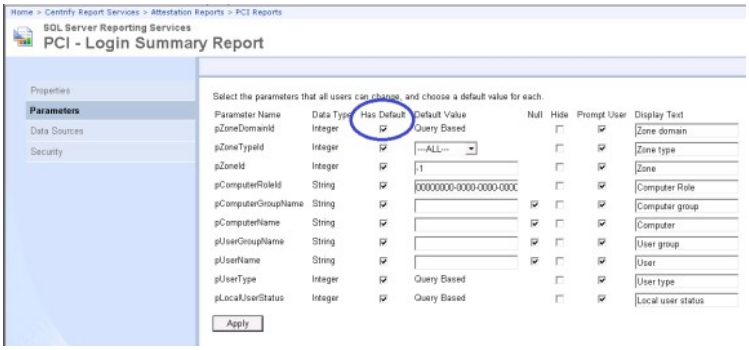
To configure a report to not run automatically when you open the report:

1. In the list of reports in the web browser, locate the desired report.
2. Move your mouse pointer over the report to open the report context menu.
3. From the context menu, select **Manage**.

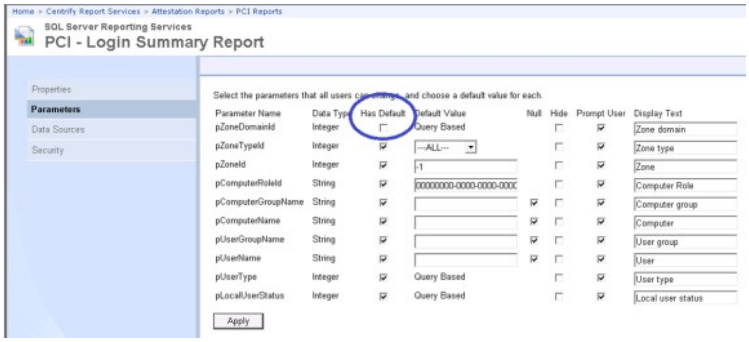


4. Select the **Parameters** page.

   Notice that 'Has Default' is selected for all parameters.

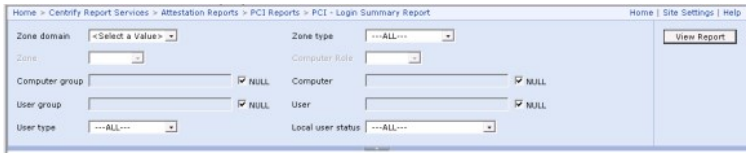


5. Deselect the 'Has Default' setting for any one parameter.
6. Click **Apply** to save the changes.



7. Open the report.

   The report does not run automatically. You can specify the filter values and click “View Report” button to run the report.



Increasing the Time-Out Value for Rebuild/Refresh Data Operations

Delinea Report Services invokes multiple database operations when it refreshes and rebuilds its cache of information stored in Active Directory. These database operations can be time-consuming in a large Active Directory environment. If any such database operation cannot be completed within a certain time period, the Delinea Report Services control panel will show that the Refresh/Rebuild process failed.

Symptom

When Delinea Report Services perform a snapshot rebuilding or refreshing and the amount of the monitored data is too large to be processed within the time-out period, this error will occur:

A database operation error occurred. Please contact your administrator to make sure the remote database is accessible and working properly. ---> System.Data.SqlClient.SqlException: Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

Resolution

You can change the time-out value (3,600 seconds by default) for that time period by performing the following steps:

1. Open the registry editor and then locate the key 'SQLCmdTimeout' under HKLM\Software\Centrify\Report Services\Service. If you cannot find it under the path, create one with the same name and as 'DWORD' type.
2. Set to 'SQLCmdTimeout' to a large enough value (unit in second) so that the rebuild/refresh/computing can be finished within the time period.

   Note: Set the SQLCmdTimeout to 0 (ZERO) mean no time-out. Customer should contact Delinea Technical Support first before changing SQLCmdTimeout to 0.

Increasing the Time-Out Values for Microsoft SQL Server Reporting Services

Consider increasing the following SSRS configuration parameter values so that the large reports can be opened successfully.

Report Execution Time-out

A report execution time-out value is the maximum number of seconds that report processing can continue before it is stopped. This value is defined at the system level. You can vary this setting for individual reports.

Symptoms

For example, you can run a report that has underlying queries that cannot be completed within the time-out period. The following error will be shown on the Report Manager like this:

An error has occurred during report processing. (rsProcessingAborted)
Query execution failed for dataset 'DataSet1'. (rsErrorExecutingCommand)
A severe error occurred on the current command. The results, if any, should be discarded. Operation cancelled by user.

Resolution

Increase the Report execution time-out value. For details, see https://msdn.microsoft.com/en-us/library/ms155782.aspx.

HTTP Runtime Execution Timeout

Symptoms

You cannot open the report and you get the following error instead. This error generally occurs when the HTTP runtime execution timeout is too short.

The remote server returned an error: (500) Internal Server Error.

Resolution

1. Open the Report Server's Web.config file, which is usually in this location:

   <Drive>:\Program Files\Microsoft SQL Server\MSRS<version number>.<instance name>\Reporting Services\ReportServer

2. Locate the HttpRuntime parameter and alter the value. If it doesn't exist, you will have to create it within the section.

```
    <trace enabled="false" requestLimit="10" pageOutpu
    <sessionState mode="Off" />
    <httpHandlers>
      <add verb="*" path="Reserved.ReportServer" type=
      <add verb="*" path="Reserved.ReportViewerWebCont
Culture=neutral, PublicKeyToken=89845dcd8080cc91" />
      <add verb="GET,HEAD" path="ScriptResource.axd" t
PublicKeyToken=31bf3856ad364e35" validate="false" />
    </httpHandlers>
    <httpModules>
      <clear />
      <add name="OutputCache" type="System.Web.Caching
      <add name="WindowsAuthentication" type="System.W
      <add name="FormsAuthentication" type="System.Web
      <add name="PassportAuthentication" type="System.
      <add name="RoleManager" type="System.Web.Securit
      <add name="UrlAuthorization" type="System.Web.Se
      <add name="FileAuthorization" type="System.Web.S
      <add name="AnonymousIdentification" type="System
      <add name="Profile" type="System.Web.Profile.Pro
      <add name="ErrorHandlerModule" type="System.Web.
/>
    </httpModules>
    <globalization requestEncoding="utf-8" responseEnc
    <httpRuntime executionTimeout="9000" />
    <securityPolicy>
      <trustLevel name="RosettaSrv" policyFile="rssrvp
    </securityPolicy>
    <trust level="RosettaSrv" originUrl="" />
    <webServices>
      <soapExtensionTypes>
        <add type="Microsoft.ReportingServices.WebServ
```

   The default value is 9000, and the value is in the seconds. The maximum value is 922337203685.

3. Increase the executionTimeout value to allow the report to be rendered.

Increasing the ReceiveTimeOut Value for Internet Explorer

Symptoms

The following error is shown when you try to open a report:

An unknown error occurred while processing the request on the server. The status code returned from the server was: 12002

Resolution

> **Note**: The resolution for this symptom involves changing a registry setting. Before you change this registry setting, you should contact Delinea Technical Support first.

You can change the ReceiveTimeout setting for Internet Explorer using the following steps:

1. Start the Windows Registry Editor.
2. Locate the following subkey:

   HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
3. In this subkey, add a ReceiveTimeout DWORD entry that has a value of (<number of seconds>)*1000.

   For example, if you want the time-out duration to be 120 minutes, set the value of the ReceiveTimeout entry to 7200000 (<120*60>*1000).
4. Restart the computer.

Using a URL to Export Report Data to CSV

Symptoms

The underlying queries in some reports take a long time to execute and you may get the following errors when opening reports:

The remote server returned an error: (500) Internal Server Error.

Resolution

Besides using the report filters to make the report take less time to execute as described in earlier section, you can export the report to CSV by using a URL. In addition, you can skip exporting the chart data for the following reports:

- PCI – Login Summary Report
- PCI – Right Summary Report
- SOX – Login Summary Report
- SOX – Right Summary Report

To configure the report URL to export to CSV and skip the chart data in the exported file:

1. Compose the URL in the following format:

   http://\<hostname\>:\<port\>/ReportServer_\<instancename\>?\<report path\>&rs:Command=Render&rs:Format=CSV&pZoneDomainId=-1&SkipChartData=True

   For example:

   This is a URL to export the PCI – Login Summary report:

   http://win2012r2/ReportServer_CENTRIFYSUITE?%2fcentrify+Report+Services%2fAttestation+Reports%2fPCI+Reports%2fPCI+-+Login+Summary+Report&rs:Command=Render&rs:Format=CSV&SkipChartData=True&pZoneDomainId=-1

   This is a URL to export the PCI – Right Summary report:

   http://win2012r2/ReportServer_CENTRIFYSUITE?%2fcentrify+Report+Services%2fAttestation+Reports%2fPCI+Reports%2fPCI+-+Right+Summary+Report&rs:Command=Render&rs:Format=CSV&SkipChartData=True&pZoneDomainId=-1
2. Access the URL in Internet Explorer.
3. Save the exported CSV file.

References

https://msdn.microsoft.com/en-us//library/ms153586.aspx

https://msdn.microsoft.com/en-us/library/ms159261.aspx

Creating the Report Subscription for CSV Export

This section shows how to use the SQL Server Reporting Services (SSRS) subscription feature to export report data to CSV regularly.

Prerequisites

- Please check whether your SQL Server edition supports the reporting subscription feature.

https://msdn.microsoft.com/en-us/library/cc645993(v=sql.100).aspx

- SQL Server Agent is already installed and running.

Configuring The Report Data Source For Subscriptions

To configure a report subscription in SSRS for CSV export and skip the chart data in the export:

1. Open Delinea Report Services.
2. Click **ReportDataSource** to open the report data source properties page.
3. Configure the report data source to store connection credentials in the report server:
    1. For the connection method, select **Credentials stored securely in the report server**.
    2. Enter the login user name and password.
    3. Select **Use as Windows credentials when connecting to the data source**.
    4. The following screenshot is an example of the connection settings configuration:

4. Secure access to the reports and the report data by adding or editing role assignments for the report folder.
    1. Open the Security page for the report folder 'Access Manager Reports' and 'Attestation Reports'.
        1. Here you can view, add, edit, or delete role assignments for the report folder.
        2. The data source uses stored credentials, which means that users who are able to view the reports would be able to read the report data. To avoid this potential risk, you can define role-based security for reports in the Security page, as shown below.

    2. Delete the default role assignment that assigns the Browser role to NT AUTHORITY\Authenticated Users to remove report read access to all authenticated users.
    3. In the report folder's Security page, click New Role Assignment.
    4. Enter the users or groups who can access the reports.

5. Select one or more roles to assign to the specified user(s).

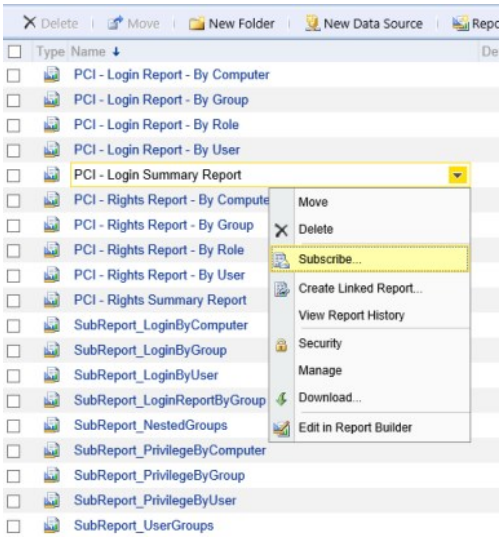
   For example, if you want the specified users to only view the report, select the Browser role.

6. Click OK to save the changes.

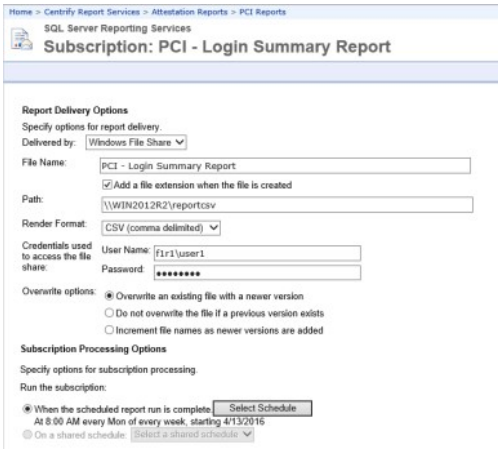
Creating A CSV Report Subscription

To configure a report subscription in SSRS for CSV export and skip the chart data in the export:

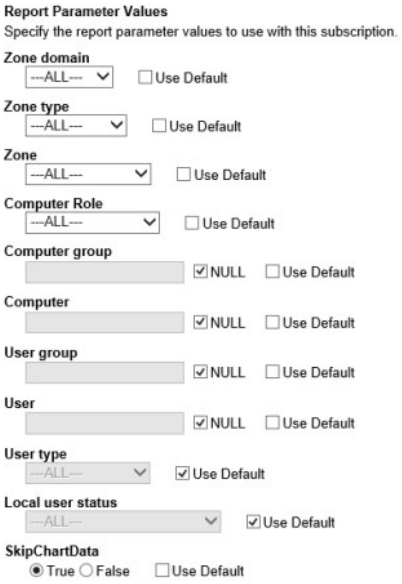
1. In the list of reports, select the report that you want to export to CSV.
2. Click the context menu and click **Subscribe**.



3. In the **Subscription** page, set the options according to the following screenshot.
   1. To specify when the scheduled report runs, click **Select Schedule**.
   2. When you specify the file path, the path must conform to the Uniform Naming Convention format.



3. In the lower area of the subscription page, set the Zone domain parameter to **ALL** in order to export report data for all zone domains.



4. After setting the options, click **OK** to create this subscription.

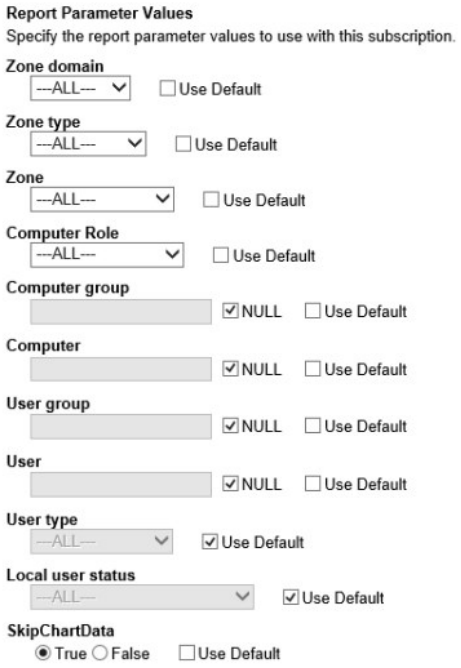
Skipping Chart Data From CSV Report Subscriptions

You can skip exporting the chart data to CSV for the following reports:

- PCI – Login Summary Report
- PCI – Right Summary Report
- SOX – Login Summary Report
- SOX – Right Summary Report

To configure a report subscription in SSRS for CSV export and skip the chart data in the export:

1. Open the report subscription. (From the report's context menu, click Manage, and then click the **Subscription** page.)
2. In the lower area of the subscription page, set the **SkipChartData** parameter to **True**.



3. After setting the options, click **OK** to save the subscription.

In general, if something doesn't work the way that you think it should, try the following to troubleshoot your reporting environment:

- View the log files
- Rebuild or refresh the reporting data
- Validate that the reporting service has the correct permissions to read data from the monitored domains and replicate the data.
- Export diagnostics data for use by Delinea Technical Support (if technical support requests that you do so).

This section describes some situations that you might encounter, along with some suggested solutions or workarounds.

You Don't See Any data When You Open a Report

Problem: You've installed everything and you can open a report, but you don't see any data.

Solution: Make sure that there has been at least one synchronization between Active Directory and the reporting database. Use the Report Configuration wizard to do this.

You Don't See the Report Builder Link in Internet Explorer

Problem: You go the Home page in Internet Explorer, the home page for your deployed reports in SSRS, and you do not see the Report Builder link. But you're fairly sure that you have the required permissions to create reports.

Solution: Here are some things for you to check:

1. Make sure that you are logging in within the same domain that SSRS is installed within. For example, if you're creating an evaluation version that uses a different domain, there may be issues.
2. Go download the Report Builder for your SQL Server version. For now, it's a separate download.

You Can't Log in to Report Services in Internet Explorer

Problem: When you log in to Delinea Report Services in Internet Explorer, you cannot successfully log in. You see an error message like this: "User domain\user does not have required permissions. Verify that sufficient permissions have been granted and Windows User Account Control (UAC) restrictions have been addressed."

Explanation: If you're seeing this issue, it may have happened after your first installation or an upgrade in which you created a new SQL Server instance.

Solution: Here are some things for you to try:

- When you go to launch Report Services, right-click it and select Run as Administrator. This may allow you to log in to Report Services, and from there you can edit the Site Settings for security.
- Log in to Report Services as an administrator, and go to Site Settings to add your users by way of adding the domain and assign the group or user a role. For details, see Granting access in SSRS to reports.
- Make sure that you also set permissions for the home folder, as mentioned in the topic mentioned above.

You Get a Server Error When You Try to Synchronize with Active Directory

Problem: In the Report Services control panel, when you go to synchronize data for report services, the following error displays: "The server is unwilling to process the request." (KB-6350)

You also see a similar error in the report services log file. Here's an example of what the error looks like:

[2015-08-21 10:53:25.714 +0800] Centrify.Report.Service.exe[3596,10] Error: SyncServer.DoSynchronization: Failure during synchronize domain a9f1r1.test, DC: a9d1-w2k12r2.a9f1r1.test.

[2015-08-21 10:53:25.714 +0800] Centrify.Report.Service.exe[3596,10] Error: SyncServer.DoSynchronization: Reason: The server is unwilling to process the request.

Explanation: The issue is due to insufficient memory on the domain controller. The domain controller is unable to allocate enough memory for Active Directory caching.

Solution: Adjust the memory allocated to the domain controller, according to Memory requirements.

Port Conflicts

Problem: Delinea Report Services not install correctly when port 80 is used by another application, such as Apache Tomcat. The following error displays during the report services configuration wizard: "The service was unable to access Report Services." (KB-7443)

Explanation: By default, Microsoft SQL Server Reporting Services (SSRS) use port 80, and it is not recommended to run it with a third party software that also uses port 80 or 443.

Solution: For port conflict situations, you can configure SSRS to use another port.

To change the port that SQL Server Reporting Services (SSRS) uses:

1. Open the SQL Server Reporting Services Configuration Manager.
2. Navigate to the **Web Service URL**.
3. Change the TCP port to an unused port other than 80.

For example, port 8080.

4. Navigate to the **Report Manager URL**, and click **Advanced**.
5. Change the TCP port to the same port number that you specified in Step 3.
6. Run the Delinea Report Services Configuration Wizard and specify URLs with the new port number.

For example, http://reportservice:8080/ReportServer_CssREPORTS2

7. Verify that you can access reports through the specified port.

You may also need to modify your firewall rules for access to the specified port.

SSRS Fails to Start on Windows 2008 R2 Systems

Problem: SQL Server Reporting Services (SSRS) fails to start, due to a timeout issue. This issue occurs only on Windows 2008 R2 systems. (KB-8065)

SSRS Produces the Following Error

Windows could not start the SQL Server Reporting Services (MSSQLSERVER) service on local computer. Error 1053: The service did not respond to the start or control request in a timely fashion.

Explanation: This happens due to SSRS checking for certificate revocation lists (CRL), and this is a Microsoft known issue, as detailed here: http://support.microsoft.com/kb/2745448.

> **Note**: Delinea does not take any responsibility for the content or availability of this link, it is provided as a courtesy. You should contact Microsoft if there are any further questions.

Solution: You can perform one of the following tasks to try and resolve this issue:

- Disable certificate revocation lists checking. For details, see http://tech.lanesnotes.com/2014/02/sql-server-reporting-services-service.html
- Change the default revocation checking behavior using group policy. For details, see https://technet.microsoft.com/en-us/library/ee619786(v=ws.10).aspx

SQL Server 2008 R2 Express Edition Produces an Installation Error

Problem: When you run the installer, you get an error when it tries to install the SQL Server 2008 R2 Express edition for report services. The installer produces the following error: (KB-8172)

The Report Services Configuration Wizard Cannot be Completed Due to an Error that Occurred

The program was unable to install SQL Server on this computer, exit code: 0x851A0017. Please refer to the Delinea Knowledge Base article (KB-4589) for more information on the error code you received. Please fix the issue and run Setup again.

You might also see something like the following errors in the SQL Server log file, which you can locate in a directory such as C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log\<number>\Detail.txt.

2017-01-25 12:41:31 Slp: Configuration action failed for feature SQL_Engine_Core_Inst during timing ConfigRC and scenario ConfigRC. 2017-01-25 12:41:31 Slp: Could not find the Database Engine startup handle. 2017-01-25 12:41:31 Slp: The configuration failure category of current exception is ConfigurationFailure 2017-01-25 12:41:31 Slp: Configuration action failed for feature SQL_Engine_Core_Inst during timing ConfigRC and scenario ConfigRC. 2017-01-25 12:41:31 Slp: Microsoft.SqlServer.Configuration.SqlEngine.SqlEngineConfigException: Could not find the Database Engine startup handle. 2017-01-25 12:41:31 Slp: at Microsoft.SqlServer.Configuration.SqlEngine.SqlServerServiceBase.WaitSqlServerStart(Process processSql) 2017-01-25 12:41:31 Slp: at Microsoft.SqlServer.Configuration.SqlEngine.SqlEngineDBStartConfig.ConfigSQLServerSystemDatabases(EffectiveProperties properties, Boolean isConfiguringTemplateDBs, Boolean useInstallInputs) 2017-01-25 12:41:31 Slp: at Microsoft.SqlServer.Configuration.SqlEngine.SqlEngineDBStartConfig.DoCommonDBStartConfig(ConfigActionTiming timing) 2017-01-25 12:41:31 Slp: at Microsoft.SqlServer.Configuration.SqlConfigBase.SlpConfigAction.ExecuteAction(String actionId) 2017-01-25 12:41:31 Slp: at Microsoft.SqlServer.Configuration.SqlConfigBase.SlpConfigAction.Execute(String actionId, TextWriter errorStream) 2017-01-25 12:41:31 Slp: Exception: Microsoft.SqlServer.Configuration.SqlEngine.SqlEngineConfigException. 2017-01-25 12:41:31 Slp: Source: Microsoft.SqlServer.Configuration.SqlServer_ConfigExtension. 2017-01-25 12:41:31 Slp: Message: Could not find the Database Engine startup handle.

Explanation: The SQL Server Express edition isn't able to use the encryption protocols provided by the server.

The installation error occurs because TLS1.0/1.1 and SSL 3.0 protocols and some ciphers have been disabled on the server, due to customer security concerns. SQL Server 2008 R2 Express Edition does not support the newer version of cipher suites (such as TLS1.2 with SHA256), while the regular versions of SQL Server with the latest updates or support packs (SP) do.

Solution: Restore the server settings back to the system defaults that allow TLS1.0/1.1, SSL 3.0 protocols and ciphers. After you do that, the installer will successfully complete a report services installation with SQL Server Express edition.

Installing SQL Server from the Delinea Management Services Installer Generates Error Codes

Problem: When you install the SQL Server version that is bundled with the Delinea Management Services installer, there are errors. (KB-4589)

0x84B40000 - full text service cannot run under local system account on DC.	This means user is trying to install SQL Server on a Domain Controller with Full Text Search service configured under a local system account. This is not supported by Microsoft. Workaround is to either install SQL on a member server, or manually install SQL and select a different account to run it as Full Text Search service.
	For this error code, more information is needed, see below on how to collect logs.
0X6AA	If SQL server is already installed on a machine with default SQL server instance (with advanced services), the SQL server setup will fail with the above error code. To workaround this issue, reinstall SQL Native Client (SNAC) before installing the second instance of SQL Server 2005 Express Edition with Advanced Services. - http://msdn.microsoft.com/en-us/sqlserver/ff658533 (Provided as a courtesy)
	This means a hyphen (-) in the SQL server's instance name has been specified and is not allowed.
	SQL Server Management Studio 2005 has been installed on the system and there is an attempt to install SQL 2008 on top of it. To workaround this issue, manually uninstall SQL Server Management Studio and then install the later version.
	PowerShell 2.0, which is a prerequisite for Microsoft SQL server 2014, is not installed. Install Windows Management Framework 2.0 first before run Report Services Configuration Wizard
	SQL Server 2008 R2 express edition with advanced features will fail to install an new instance when TLS 1.0/1.1 and SSL 3.0 protocols are disabled. It fails with a message like: SQL Server installation failed. To continue, investigate the reason for the failure, correct the problem, uninstall SQL Server, and then rerun SQL Server Setup. To work around this issue, re-enable the protocols (Update corresponding values to 1.) The SQL Server instance then can be installed successfully. Refer to: https://blogs.msdn.microsoft.com/friis/2016/07/25/disabling-tls-1-0-on-your-windows-2008-r2-server-just-because-you-still-have-one/

If the SQL Server installation fails for any other reason, send the installation log files to Delinea support. You can locate the installation log files in the following locations:

- SQL Server 2008 and 2008 R2:

%ProgramFiles%\Microsoft SQL Server\100\Setup Bootstrap\LOG

- SQL Server 2012:

%ProgramFiles%\Microsoft SQL Server\110\Setup Bootstrap\Log\.

- SQL Server 2014:

%ProgramFiles%\MicrosoftSQL Server\120\Setup Bootstrap\Log\.

- SQL Server 2016:

%ProgramFiles%\ Microsoft SQL Server\130\Setup Bootstrap\Log\

See also: https://centrify.my.salesforce.com/50180000000blYD, http://support.microsoft.com/kb/955396.

Can't install SQL Server 2012 or 2014 instance on Windows 2008 SP2

Problem: If you use the Report Services Configuration wizard to install a new instance of SQL Server version 2012 or 2014 on Windows Server 2008 SP2, the installation fails if Windows Powershell 2.0+ or Windows Management Framework 2.0 is not already installed. The installation failure has an exit code of 0x84BE0260 (KB-7096).

Explanation: Windows Server 2008 SP2 doesn't include PowerShell 2.0 or Windows Management Framework 2.0 by default. Later versions of Windows Server do include these components by default.

Solution: Install PowerShell 2.0 or higher and Windows Management Framework 2.0 before you run the Report Services Configuration Wizard to install a new instance of Microsoft SQL Server 2012 or 2014.

You can download Windows Management Framework 2.0 from https://support.microsoft.com/en-us/kb/968930.

Report Services computation takes longer than it used to

Problem: If Report Services uses SQL Server 2014 or above, you might notice that Report Services spends more time on computation.

Explanation: In SQL Server 2014, Microsoft introduced a new cardinality estimator (CE). This cardinality estimator was redesigned to improve query performance, and there may be some performance degradation for some SQL statements.

Solution: If you notice some report services computation performance issues, set the database compatibility level to 110 to force SQL Server to use the old cardinality estimator.

To set the database compatibility level to 110:

1. In SQL Server Management studio, run the following before Report Services synchronizes with Active Directory:

ALTER DATABASE <the database name deployed by Report Services> SET COMPATIBILITY_LEVEL = 110

Frequently asked questions about report services

Question: Is it possible for report services to use an existing database that's already been created according to our organization's standards?

Answer: Report services cannot use an existing database, the Configuration Wizard creates a new database.

Question: Does report services create just one database?

Answer: Yes. If you reconfigure report services, the Configuration Wizard creates a new database.

Question: Does the report services installation make any other modifications to database objects other than in the database it creates?

Answer: No.

Delinea

This section covers which Active Directory attributes that report services synchronizes for use in reports. Report services synchronizes these attributes from Active Directory to the reports database in a one-way synchronization process.

AD Computer

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name sAMAccountName userAccountControl primaryGroupID dNSHostName operatingSystem operatingSystemVersion operatingSystemServicePack description whenCreated pwdLastSet objectSid sIDHistory managedBy location givenName postalAddress

AD Group

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description gidNumber groupType mail member msSFU30GidNumber msSFU30Name msSFU30NisDomain objectSid primaryGroupToken sAMAccountName sIDHistory whenCreated managedBy

AD User

| Active Directory class | user |

| Available in zone mode? | Yes | | Available in domain mode? | Yes | | Attributes | objectGUID parentGUID name sAMAccountName userPrincipalName userAccountControl primaryGroupID msSFU30NisDomain uid uidNumber | | Attributes (continued) | gidNumber loginShell unixHomeDirectory gecos msSFU30Name msSFUUidNumber msSFU30GidNumber msSFU30HomeDirectory msSFU30Gecos whenCreated | | Attributes (continued) | lastLogonTimestamp accountExpires lockoutTime pwdLastSet givenName sn initials displayName description | | Attributes (continued) | physicalDeliveryOfficeName telephoneNumber mail wWWHomePage objectSid SIDHistory streetAddress postOfficeBox | st | | Attributes (continued) | postalCode co homePhone otherHomePhone pager otherPager mobile otherMobile facsimileTelephoneNumber otherFacsimileTelephoneNumber | | Attributes (continued) | ipPhone otherIpPhone title department company manager profilePath scriptPath homeDirectory homeDrive msNPAllowDialin | | Attributes (continued) | msNPCallingStationID msRADIUSServiceType msRADIUSCallbackNumber msRADIUSFramedIPAddress msRADIUSFramedRoute |

Application Right

Available in zone mode?	Yes

Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Command Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Computer Role

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Computer SCP

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName keywords managedBy whenCreated

Computer Zone AzScope

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName msDS-AzScopeName

Computer Zone Container

Available in zone mode?	Yes
Available in domain mode?	Yes

Attributes	objectGUID parentGUID name displayName description

Container

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name

Desktop Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Domain

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	msDS-LogonTimeSyncInterval distinguishedName lockoutDuration

Dzsh Command Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Group SCP

Available in zone mode?	Yes
Available in domain mode?	Yes

Attributes	objectGUID parentGUID name displayName keywords gidNumber managedBy

License Container

Available in zone mode?	No
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName description whenCreated

Local Group SCP

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName keywords gidNumber

Local User SCP

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName keywords uid uidNumber gidNumber unixHomeDirectory loginShell gecos

Network Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Pam Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData

Privileged Command Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-AzOperationID

Restricted Environment

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-OperationsForAzTask

Role

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData msDS-OperationsForAzTask msDS-TasksForAzTask

Role Assignment

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName msDS-AzApplicationData msDS-TasksForAzRole msDS-MembersForAzRole

Ssh Right

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description msDS-AzApplicationData

User SCP

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name displayName name keywords uid uidNumber gidNumber unixHomeDirectory loginShell gecos managedBy

Zone

Available in zone mode?	Yes
Available in domain mode?	Yes
Attributes	objectGUID parentGUID name description displayName

Delinea

Auditing Guides

- Auditing Administrator Guide
- Audit Database Management Guide
- Find Sessions Guide

Overview of the Auditing Infrastructure

Auditing is a key feature of Server Suite. If you choose to enable auditing in your organization, you can capture detailed information about user activity on Linux, UNIX, and Windows computers and store that activity to improve regulatory compliance and accountability and mitigate security risks. This section provides an overview of the auditing infrastructure, including key components and terminology.

The following topics are covered:

Delinea

Deciding Whether to Audit User Activity

Just as it is important to protect assets and resources from unauthorized access, it is equally important to track what users who have permission to access those resources are doing or have done in the past. For users who have privileged access to computers and applications with sensitive information, auditing their actions helps ensure accountability and improve regulatory compliance.

There are many reasons for organizations to establish auditing policies and enable auditing of user activity. For example, you might want to audit activity for any of the following reasons:

- To prove certain computers or applications are secure in order to comply with government or industry regulatory requirements.
- To report on actions taken by users with elevated privileges.
- To prevent the use of shared passwords when more than one person needs administrative access to a computer or an application.
- To improve accountability when users with elevated permissions have access to privileged resources.
- To detect suspicious activity and mitigate the threat posed by malicious insiders or third parties who have access to sensitive systems.
- To pinpoint actions that may have caused failures and simplify troubleshooting procedures.
- To capture information, such as the steps that resolved an open case, that can be used to help your organization improve its helpdesk operations or security procedures.

Capturing Detailed and Summary Information for User Sessions

After you deploy the auditing infrastructure, you can capture detailed information about user activity and the events that occurred on the computers you choose to audit. On those computers, an agent starts recording user activity when a user selects an audited role or starts a login shell locally, using a remote shell, or through a virtual network connection such as Citrix or VNC.

Each record of continuous user activity is called a **session**. A session ends when the user logs out, disconnects, or is inactive long enough to lock the desktop. If the user reconnects or unlocks the desktop, the agent resumes recording the user's activity as a new session. When users start a new session on an audited computer, they can be notified that their session is being audited but they cannot turn off auditing except by logging off, so you have a complete record of what happened, includes an audit trail of the actions a user has taken.

You can choose whether to record only summaries of user activity or a full visual record of user activity.

Sessions include different kinds of information depending on the audited system's operating system:

- **Windows**: When auditing Windows computers, each session is a video capture of everything that takes place on the desktop, including the applications opened, text that was entered, and the results that were displayed.
- **Linux:** When auditing Linux computers, the agent records shell activity, such as the commands a user runs or the changes made to key files and data. On some versions of Linux computers, actions performed using a display manager, such as GNOME or KDE, are also recorded. Consult the Delinea release notes for supported platform details.

In addition to capturing detailed information about user activity, sessions provide a summary of actions taken so that you can scan the applications opened or commands executed for potentially interesting or damaging actions without playing back a complete session. After you select a session of interest in the Audit Analyzer, the console displays an indexed list of actions taken in the order in which they occurred. You can then select any entry in the list to start viewing the session beginning with that action. For example, if a user opened an application that stores credit card information, you can scan the list of actions for that event and begin reviewing what happened in the session from the time the user opened that particular application.

If users change their account permissions to take any action with elevated privileges, the change is recorded as an audit trail event. You can also search for these events to find sessions of interest.

Reviewing Recorded Activity

The information recorded in each session is transferred to a Microsoft SQL Server database so that it is available for querying and playback. Because the information is collected as it happens, you can monitor computers for suspicious activity or troubleshoot problems immediately after they occur.

You can also search for and play back sessions to locate past events that occurred on specific computers or that affected particular users. For example, you might be interested in activity that occurred immediately before a security breach or want to investigate the cause of an application failure. Similarly, a security expert might want to see who had access to computers with sensitive data, such as payroll information or medical records, during a particular period of time, such as the last 72 hours.

Auditing Requires a Scalable Architecture

To ensure scalability for large organizations and provide fault tolerance, the auditing infrastructure has a multi-tier architecture that consists of the following layers:

- **Audited computers** are the computers on which you want to monitor activity. To be audited, the computer must have an agent installed, audit features enabled, and be joined to an Active Directory domain.
- **Collectors** are intermediate services that receive and compress the captured activity from the agents on audited computers as the activity occurs. You should establish at least two collectors to ensure that auditing is not interrupted. You can add collectors to your installation at any time and it is common to have multiple collectors to provide load balancing and redundancy.
- **Audit stores** define a scope for auditing and include the audit store databases that receive captured activity and audit trail records from the collectors and store it for querying and playback. Audit store databases also keep track of all the agents and collectors you deploy. For scalability and network efficiency, you can have multiple audit stores each with multiple databases.
- A **management database server** is a computer that hosts the Microsoft SQL Server instance with the audit management database. The management database stores information about the overall installation, such as the scope of each audit store, which audit store database is active and where there are attached databases, the audit roles you create, and the permissions you define. The management database enables centralized monitoring and reporting across all audit stores, collectors, and audited computers.
- The **Audit Manager** and **Audit Analyzer consoles** are the graphical user interfaces which administrators can use to configure and manage the deployment of audit components, such as agents and collectors, or to query and review captured user sessions.

To ensure that audit data transferred over the network is secure, communication between components is authenticated and encrypted.

In addition to these core components of the auditing infrastructure, there is a separate Windows service that collects audit trail events when there are audit store databases that are not accessible, for example, because of network issues or the database server is shut down. This audit management server runs as a Windows service and spools the events on the management database, then sends them to the audit store database when the inaccessible database comes back online.

In addition to spooling audit trail events, the audit management server automatically calculates the approximate disk space used by audited sessions on the database server. The audit management server will calculate the session size for all completed audited sessions. The session size is not calculated for in-progress or disconnected sessions. You can view the session size for all completed sessions in the Audit Analyzer console's query results.

How Audited Sessions are Collected and Stored

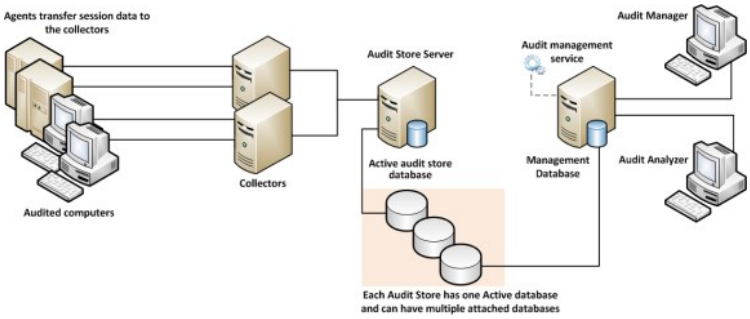
The agent on each audited computer captures user activity and forwards it to a collector on a Windows computer. If the agent cannot connect to a collector—for example, because all of the computers hosting the collector service for the agent are shut down for maintenance—the agent spools the session data locally and transfers it to a collector later.

The collector sends the data to an audit store server, where the audit data is stored in the Microsoft SQL Server database that you have designated as the **active audit store database**. As you accumulate data, you can add more SQL Server databases to the audit store to hold historical information or to change the database designated as the active audit store database.

After the audit data is transferred to the audit store database, you can use the Audit Analyzer console to request session data. The audit management database, which stores information about all of the components that make up the auditing infrastructure, retrieves the session data from the appropriate audit store database.

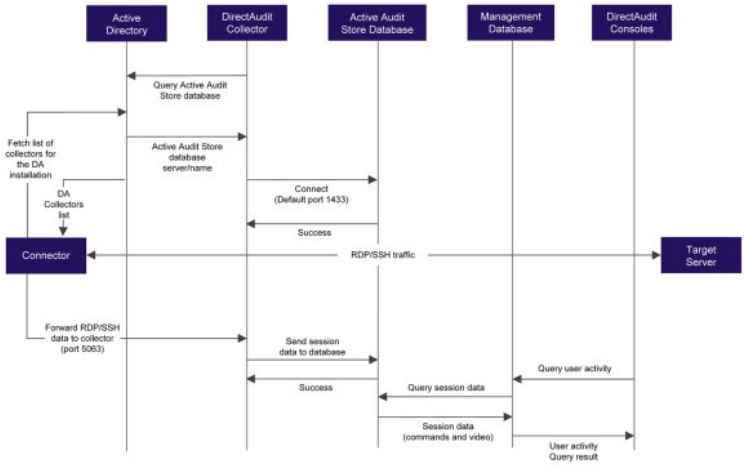
Auditing Architecture and Dataflow

The following figure illustrates the basic architecture and flow of data with a minimum number of auditing components installed.

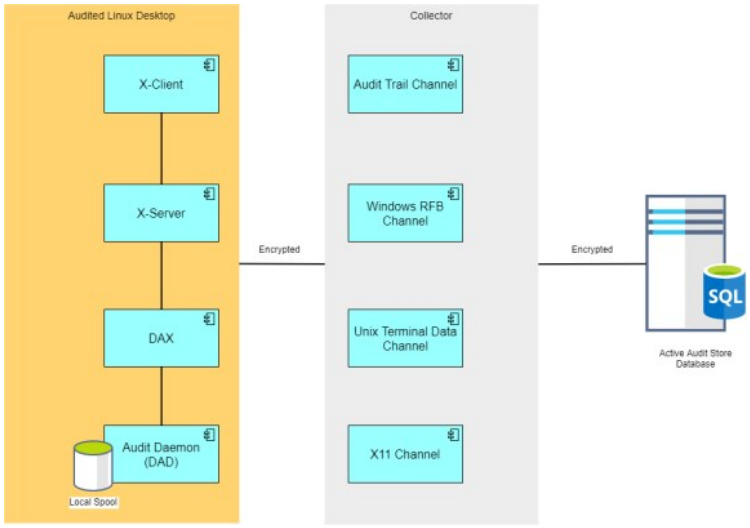


In the illustration, each agent connects to one collector. In a production environment, you can configure agents to allow connections to additional collectors for redundancy and load balancing or to prevent connections between specific agents and collectors. You can also add audit stores and configure which connections are allowed or restricted. The size and complexity of the auditing infrastructure depends on how you want to optimize your network topology, how many computers you are auditing, how much audit data you want to collect and store, and how long you plan to retain audit records.

The following figure illustrates the data flow details. You can see which components communicate to other components and in what order. The diagram also includes some port details.



The following diagram shows how the Linux Desktop auditing session data is collected.



Within the Linux Desktop, there's a component called DAX that generates the recorded session data and passes it to the audit daemon. The audit daemon encrypts and passes the recorded session data to the collector. The collector channels session data of different types together and passes that encrypted session data along to the active audit store database.

Deploying Auditing Components in an Audit Installation

The multi-tiered architecture of the auditing infrastructure is referred to collectively as a **DirectAudit installation**. The DirectAudit installation represents a logical object similar to an Active Directory forest or site. It encompasses all of the auditing components you deploy—agents, collectors, audit stores, management database, and consoles—regardless of how they are distributed on your network. The installation also defines the scope of audit data available. All queries and reports are against the audit data contained within the installation boundary.

The most common deployment scenario is to have a single audit installation for an entire organization so that all audit data and management of the audit data is centralized. Within a single installation, you can have components wherever they are needed, as long as you have the appropriate network connections that allow them to communicate with each other. The audit data for the entire installation is available to users who have permission to query and view it using a console. For most organizations, having a single installation is a scalable solution that allows a "separation of duties" security model through the use of audit roles. If you establish a single installation, there will be one Master Auditor role for the entire organization, and that Master Auditor can control the audit data that other users and groups can see or respond to by defining roles that limit access rights and privileges.

However, if you have different lines of business with different audit policies—in different geographic locations, or with different administrative groups—you can configure them as separate audit installations. For example, if you have offices in North America and Hong Kong managed by two different IT teams—IT-US and IT-HK—you might want to create two DirectAudit installations to maintain your existing separation of duties for the ITUS and IT-HK teams.

Planning Where to Install Auditing Components

Before you install Centrify Audit & Monitoring Service, you should develop a basic deployment plan for how you will distribute and manage the components that make up an installation. For example, you should decide how many collectors and audit stores to create and where to put them. You should also consider the network connections required and how many computers you plan to audit. For example, you can have multiple agents using the same set of collectors, but you should keep the collectors within one hop of the agents they serve and within one hop of the audit stores to which they transfer data.

By planning where to install components initially, you can determine the number of collectors you should have for load balancing or redundancy. After the initial deployment, you can add collectors and audit stores whenever and wherever they are needed.

Using Multiple Databases in an Audit Store

Each audit store uses Microsoft SQL Server to provide database services to the audit installation. When you install the first audit store, you configure the database instance you want to use and that database becomes the active database for storing incoming audit data. A single audit store, however, can have several databases attached to it. Attached databases store historical information and respond to queries from the management database. You can use the Audit Manager console to control the databases that are attached to the audit store and to designate which database is active. Only one database can be active in an audit store at any given time.

Although the audit store can use multiple databases, the presentation of session data is not affected. If a session spans two or more databases that are attached to the audit store, the Audit Analyzer console presents the data as a single, unbroken session. For example, if you change the active database during a session, some of the session data is stored in the attached database that is no longer active and some of it stored in the newly activated database, but the session data plays back as a single session to the auditor.

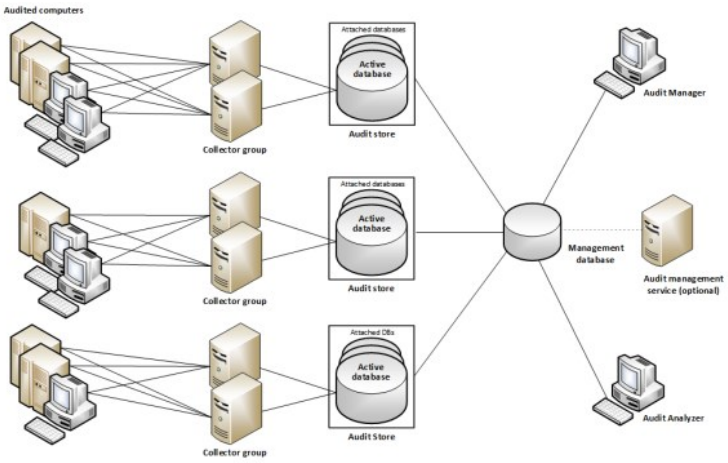
Using Multiple Consoles in an Installation

A single installation always has a single audit management database. In most cases, however, you use more than one console to request data from the audit management database. The two most important consoles in an installation are the Audit Manager console and the Audit Analyzer console.

- As the audit installation owner, you use the Audit Manager console to configure and manage the auditing components in your installation. In most organizations, there is only one Audit Manager console installed.
- Auditors use the Audit Analyzer console to search, retrieve, and play back sessions. The auditor can use predefined queries to find sessions or define new queries. Auditors can also choose whether to share their queries with other auditors or keep them private. In most organizations, there are multiple Audit Analyzer consoles installed.

In addition to the Audit Manager and Audit Analyzer consoles, you can use the Agent Control Panel and the Collector Control Panel to configure and manage agents and collectors.

The following figure shows the architecture of a medium-size installation.



Audit Installation

Agent Components

On Audited UNIX computers

To enable auditing for Linux and UNIX computers, you must install the Centrify Agent for *NIX on the computers you want to audit and make sure the computers are joined to an Active Directory domain. Joining a domain is required to ensure that authentication and authorization services are provided by Active Directory. To enable auditing on a computer, the Centrify Agent for *NIX includes the following components:

- dad—the core auditing service that collects the audit data and either sends it to a collector or spools it locally until a collector is available.
- cdash—the UNIX shell wrapper that intercepts all user traffic and sends it to the dad process.
- dacontrol, dainfo, dareload, and other command-line programs that enable you to manage agent operations from a login shell.
- dax—the audit service that records graphical user interface sessions on xWindows computers. Consult the release notes for which xWindows versions are supported.

If you're auditing only shell sessions on a UNIX computer: after you enable auditing on a computer, the agent captures all output (stdout), error messages (stderr), and user input (stdin) except for passwords. By default, the agent captures user input even if a user runs commands with echo turned off. For example, if a user logs on, then runs echo off before typing the sudo command, the auditing service captures the sudo entry as part of the user's session.

If you're auditing xWindows sessions: the agent captures all windows that a user opens and which user interface items the user interacts with. For web browser applications, the agent captures the title of the web page but not any activity within the web page.

On Audited Windows Computers

To enable auditing for Windows computers, you must install the Centrify Agent for Windows on the computers you want to audit and make sure the computers are joined to an Active Directory domain. Joining a domain is required to ensure that authentication and authorization services are provided by Active Directory. If you enable auditing for the Centrify Agent for Windows, the agent includes the following components:

- wdad—the Windows audit data collection service.
- wash—the Windows service that intercepts all user traffic and sends it to the Windows audit data collection service.
- The Agent Control Panel—an applet that enables you to configure and manage the agent.

For example, you can use the Agent Control Panel to configure the color depth of audit data to achieve the desired balance between playback screen resolution and audit store database size.

Planning an Audit Installation

This section describes the decisions you need to make during the planning phase of a deployment and summarizes what's involved in deploying audit and monitoring service components and auditing-related services on the computers to be audited. It includes simplified diagrams that highlight the steps involved.

The following topics are covered:

- Deciding on the scope of the installation
- Deciding where to install the management database
- Deciding where to install collectors and audit stores
- Deciding where to install agents
- Deciding where to install consoles

- Supported SQL Server editions
- Checking SQL Server logins for auditing
- Determining storage requirements for auditing
- What's involved in the deployment process

Deciding on the Scope of the Installation

Before you deploy any part of the auditing infrastructure, you should decide on the scope of the audit installation and whether you want to use a single installation for your entire Active Directory site, or separate audit installations for different geographical areas or functional groups.

The most common deployment scenario is a single installation for each Active Directory forest, so that auditors can query and review information for the entire organization. However, if your Active Directory site has more than one forest, you might want to use more than one installation. If you want to use more than one installation, you should determine the subnetwork segments that will define the scope of each installation.

In Active Directory, a site represents the collection of Internet Protocol (IP) addresses that describe the physical structure of your network. If you are not familiar with how Active Directory sites are defined, you should consult Microsoft documentation for more information.

Delinea

Deciding Where to Install Different Audit Components

It is important to plan the installation of all the different audit and monitoring services components.

- Deciding where to install the management database
- Deciding where to install collectors and audit stores
- Deciding where to install agents
- Deciding where to install consoles

Deciding Where to Install the Management DDatabase

Each audit installation has a single audit management server and audit management database. The management database is a Microsoft SQL Server database that stores information about the installation such as the Active Directory sites or subnets associated with each audit store.

The computer you use for the audit management database should have reliable, high-speed network connectivity. The management database does not store the captured sessions, and is, therefore, much smaller than the audit store databases. There are no specific sizing requirements or recommendations for the management database.

You can use the following guideline as the recommended minimum hardware configuration for the computer you use as the management database:

Management database	Any	1 to 2	2.33 GHz	8 GB

The audit management server is a Windows service that performs two main tasks:

- The service collects audit trail events on the management database, then sends them to the audit store database.
- The service automatically calculates the approximate disk space used by audited sessions.

Deciding Where to Install Collectors and Audit Stores

Although a collector and an audit store database can be installed on the same computer for evaluation, you should avoid doing so in a production environment. As part of the planning process, therefore, you need to decide where to install collectors and audit store databases. In designing the network topology for the installation, there are several factors to consider. For example, you should consider the following:

- Database load and capacity
- Network connectivity
- Port requirements
- Active Directory requirements

The next sections provide guidelines and recommendations to help you decide where to install the collectors and audit store databases required to support the number of computers you plan to audit.

Use Separate Computers for Collectors and Audit Store Databases

To avoid overloading the computers that host collectors and audit store databases, you should install collectors and audit store SQL Server databases on separate computers. Because SQL Server uses physical memory to store database information for fast query results, you should use a dedicated computer for the audit store database, and allocate up to 80% of the computer's memory to SQL Server. In most installations, you also need to plan for more than one audit store database and to periodically rotate from one database to another to prevent any one database from getting too large. For more information about managing audit store databases, see Managing audit store databases.

Plan for Network Traffic and Default Ports

You should minimize the distance network packets have to travel between an agent and its collector. You should also minimize the distance between collectors and their audit stores. If possible, you should not have more than one gateway or router hop between an agent and its collector.

To help you plan for network traffic, the following ports are used in the initial set of network transactions:

- Directory Service - Global Catalog lookup request on port 3268.
- Authentication Services - LDAP sealed request on port 389.
- Kerberos - Ticket Granting Ticket (TGT) request on port 88.
- Network Time Protocol (NTP) Server - Time synchronized for Kerberos on port 123.
- Domain Name Service (DNS) - Host (A), Pointer (PTR), Service Location (SRV) records on port 53.

Depending on the specific components you deploy and operations performed, you might need to open additional ports. The following table summarizes the ports used for Server Suite software.

23	TCP communication for Telnet connections	Server Suite authentication service, privilege elevation service, and audit and monitoring service. By default, telnet connections are not allowed because passwords are transferred over the network as plain text.
53	TCP/UDP communication	Clients use the Active Directory DNS server for DNS lookup requests.
88	Encrypted UDP communication	Kerberos ticket validation and authentication, agents, Server Suite PuTTY
123	UDP communication for simple network time protocol (NTP)	Keeps time synchronized between clients and Active Directory for Kerberos ticketing.
389	Encrypted TCP/UDP communication	Active Directory authentication and client LDAP service.
443	Cloud Connector communication with Privileged Access Service	Cloud Connector
445	Encrypted TCP/UDP communication for delivery of group policies	The adclient and adgpupdate use Samba (SMB) and Windows file sharing to download and update group policies, if applicable.

464	Encrypted TCP/UDP communication for Kerberos password changes	Kerberos ticket validation and authentication for agents, Server Suite PuTTY, adpasswd, and passwd.
1433	Encrypted TCP communication for the collector connection to Microsoft SQL Server	The collector service sends audited activity to the database
3268	Encrypted TCP communication	Active Directory authentication and LDAP global catalog updates.
5063	Encrypted TCP/RPC communication for the agent connection to collectors	The auditing service records user activity on an audited computer.
5064	Encrypted SSL/TLS communication for the agent connection to collectors for systems that are not joined to Active Directory.	The auditing service records user activity on an audited computer outside of Active Directory.
none	ICMP (ping) connections	To determine whether if a remote computer is reachable.

Identify an Active Directory Site or Subnets

Depending on the size and distribution of your Active Directory site, an audit store might cover an entire site or specific subnet segments. If you have a large, widely distributed site, you should consider network connectivity and latency issues in determining which subnets each audit store should serve. In addition, you should always place collectors in the same site as the agents from which they receive data. Collectors and agents must always be in the same Active Directory forest. If possible, you should put collectors and agents in the same domain.

> **Note**: If you deploy agents in a perimeter network, such as a demilitarized zone (DMZ), that is separated from your main network by a firewall, put the collectors in the same Active Directory domain as the audited computers. The collectors can communicate with the audit store database through a firewall.

Determining How Many Collectors and Audit Stores to Install

Although you can add collectors and audit stores to your audit installation after the initial deployment, you might want to calculate how many you will need before you begin deploying components. You should always have at least two collectors to provide redundancy. As you increase the number of agents deployed, you should consider adding collectors.

Estimate the Number of Agents and Sessions Audited

If you plan to use more than the minimum number of collectors, the most important factor to consider is the number of concurrent sessions you expect to monitor on audited computers. The number of concurrent sessions represents the number of agents that are actively capturing user sessions in a site at the same time.

Guidelines for Linux and UNIX Computers

You can use the following guidelines as a starting point and adjust after you have observed how much audit data you are collecting and storing for Linux and UNIX computers:

500 (or less) agents	2	1
up to 1000 agents	4	1
more than 1000 agents	2 for every 500 agents	1 for every 1000 agents

Guidelines for Windows Computers or Mixed Environments

You can use the following guidelines as a starting point and adjust after you have observed how much audit data you are collecting and storing for Windows computers:

100 (or less) agents	2	1
more than 100 agents	2 for every 100 agents	1 for every 100 agents

If you auditing Linux, UNIX, and Windows computers, use the numbers of collectors and audit stores recommended for Windows agents unless you have significantly fewer Windows agents.

Determine the Recommended Hardware Configuration

The hardware requirements for collectors and audit store servers depend on the size of the installation and where the components are installed on the network. For example, the requirements for a computer that hosts the collector service are determined by the number of audited computers the collector supports, the level of user activity being captured and transferred, and the speed of the network connection between the agents and the collector and between the collector and its audit store.

Guidelines for Linux and UNIX Computers

You can use the following guidelines as the recommended hardware configuration for the computers you use for collectors and audit store servers when auditing Linux and UNIX computers:

Collectors	Up to 250 active UNIX agents	2	2.33 GHz	8 GB
	250 to 500 active UNIX agents	4	2.33 GHz	16 GB
Audit store	Up to 250 active UNIX agents	2	2.33 GHz	8 GB
	250 to 500 active UNIX agents	4	2.33 GHz	16 GB
	500 to 1000 active UNIX agents	4	2.33 GHz	32 GB

Guidelines for Windows Computers

You can use the following guidelines as the recommended hardware configuration for the computers you use as collectors and audit store servers when auditing Windows computers:

Collectors	Up to 100 active Windows agents	2	2.33 GHz	8 GB
Audit store	Up to 200 active Windows agents	2	2.33 GHz	8 GB
	200 to 500 active Windows agents	4	2.33 GHz	32 GB

Guidelines for Storage

Because audit and monitoring service collectors send captured user sessions to the active SQL Server database, you should optimize SQL Server storage for fast data logging, if possible. For the active database, you get the most benefit from improvements to disk write performance. Read performance is secondary. Fibre Attached Storage (FAS) and Storage Area Network (SAN) solutions can provide 2 to 10 times better performance than Direct Attached Storage (DAS), but at a higher cost. For attached databases that are only used to store information for queries, you can use lower-cost storage options.

Guidelines for Disk Layout

The following table outlines the recommended disk arrays:

Operating system	C: RAID 1	Operating system files, page file, and SQL Server binaries.
Microsoft SQL Server	D: RAID 10 (1+0)	Audit store database.
	E: RAID 10 (1+0)	Audit store database log files.
	F: RAID 1 or 10 (1+0)	Temporary database space (tempdb) for large queries for reports.
	G: RAID 1	Database dump files.

The size of disk needed depends on the number, length, and types of sessions recorded each day, the selected recovery model, and your data retention policies. For more information about managing audit store databases, see Managing audit store databases.

Deciding where to install agents

The Centrify Agent must be installed on all of the computers you want to audit. Therefore, as part of your planning process, you should decide whether you want to audit every computer on the network or specific computers, such as the computers used as servers or used to run administrative software.

Before installing the Centrify Agent for Windows, verify the following:

- The computer is joined to Active Directory.
- The computer has .NET 4.6.2 or later installed.
- The computer has Microsoft Windows Installer version 3.1 or newer.

Agents can communicate with a collector only if the agents and collector are in the same Active Directory forest.

For UNIX and Linux systems, be aware that desktop auditing is available only for some Linux distributions. Please see the release notes for the supported platform details.

Linux desktop auditing workings independently from shell session auditing. For platforms that support both, you can enable either one or both.

Deciding where to install consoles

You can install and run the Audit Manager console and the Audit Analyzer console on the same computer or on different computers. The computers where you install the consoles must be joined to the Active Directory domain and be able to access the management database that serves the installation.

You can also use the Audit Analyzer console to run queries from any additional computers with network access to the management database. Therefore, you should decide where it would be convenient to have this capability.

Delinea

Audit and Monitoring Deployment Checklist

The following checklist provides an overview of each of the main steps that are involved when you deploy the Audit & Monitoring Service. For any tasks related to other Delinea software, there are links to more information and procedures.

For authentication and privilege elevation deployment steps, please see Authentication and Privilege Elevation services deployment checklist.

	Preparation and Planning		
1	Analyze your network topology to determine where to install components and services and any hardware or software updates required.		Overview of the Auditing Infrastructure
2	Create a list of the computers where you plan to install different components.		Planning an Audit Installation
3	Determine the scope of the audit installation.		Deciding on the Scope of the Installation
4	Determine the size of your database storage.		Sizing Recommendations for Audit Installations
	Pre-Requisite Tasks		
5	Create Active Directory security groups for managing the permissions required for the auditing and monitoring service infrastructure.		Creating Security Groups for Auditing
6	Install Microsoft SQL Server and create a database instance for use with the audit and monitoring service.		Installing and Configuring Microsoft SQL Server for Auditing
7	Prepare SQL Server for auditing.	This includes creating a backend service account that will run stored procedures.	Configuring SQL Server to Prepare for Auditing
8	Create a setup user account and give it database administrator (DBA) privileges.	You'll use this account and password to run the installers.	Creating a Setup User Account for Installation
	Install Tasks		
9	Install the Audit Manager and Audit Analyzer consoles.		Installing the Audit Manager and Audit Analyzer Consoles
10	In Audit Manager, create a new installation for auditing.		Creating a New Installation
11	In Audit Manager, set up the Audit Stores and Audit Store databases.		Creating the First Audit Store, Creating the First Audit Store Database
12	Install and configure the audit collector service on at least two Windows computers.		Installing the Audit Collectors
13	Install a Server Suite Agent for Windows on each Windows computer that you want to audit.		Installing the Server Suite Agent for Windows
14	Install a Server Suite Agent for *NIX on each UNIX or Linux computer that you want to audit.		Installing an Server Suite Agent for

15	Install and configure the Audit Management Server component on a Windows server computer.	For this task, run the installer using the setup user account that you created in step 8.	Installing the Audit Management Server and Configuring the Audit Management Server
16	Configure and enable auditing on the Windows computers, if they're not already enabled.		Enabling or Disabling Auditing on Windows Computers
17	Configure and enable auditing on the UNIX or Linux computers.		Enabling or Disabling Auditing on Linux and UNIX Computers
18	Install additional Audit Manager or Audit Analyzer consoles on any Windows computer that you want to use for the auditing and monitoring service.		Installing Additional Audit Manager or Audit Analyzer Consoles
	Verification Tasks		
19	Verify that data is being collected and agents are working correctly: Run dainfo on audited UNIX computers. Use Audit Analyzer to verify that data is being collected.		Checking the Status of the UNIX Agent

Supported SQL Server Editions

The current release of the Audit & Monitoring Service supports 64-bit versions of the following SQL Server editions:

- SQL Server 2008 Express with Advanced Services
- SQL Server 2008
- SQL Server 2008 R2 Express with Advanced Services (Service Pack 2 or higher recommended)
- SQL Server 2008 R2 (Service Pack 2 or higher recommended)
- SQL Server 2012 Express with Advanced Services
- SQL Server 2012 (All SP levels)
- SQL Server 2014 Express with Advanced Services
- SQL Server 2014 (All SP levels)
- SQL Server 2016 -- all SP levels for SQL Server 2016 Standard and Enterprise including the latest 2016 SP2 CU7 version.
- SQL Server 2017
- SQL Server 2017 Express Advanced
- SQL Server 2019
- SQL Server 2019 Express Advanced

  Note: SQL Server 2008 and 2008 R2 are not compatible with Windows 10

Checking SQL Server Logins for Auditing

An audit installation requires at least two Microsoft SQL Server databases: one for the management database and at least one for the first audit store database. To successfully connect to these databases, you must ensure that the appropriate users and computers have permission to read or to read and write for the databases that store audit-related information.

The simplest way to manage SQL Server logins for auditors and administrators is to do the following:

- Ensure you have a SQL Server login account for the NT Authority\System built-in account.
- Add the NT Authority\System account to the sysadmin fixed server role.
- Use the Audit Manager console to add Active Directory users and groups to the Auditor roles and/or assign them administrative rights over the audit installation.

If you use Audit Manager to manage SQL Server logins, you can use Active Directory membership to automatically add and remove the permissions required for auditing activity. There is no requirement to use the SQL Server Management Studio to manage logins or permissions. Since it is recommended that you have a dedicated SQL Server instance for auditing, giving the NT Authority\System account a SQL Server login and system administrator role is an acceptable solution for most organizations.

Auditing Permissions for SQL Server

NT Authority\System	machine account	SQL Server Roles: sysadmin role

Creating Security Groups for Auditing

Depending on whether you configure Microsoft SQL Server to use Windows only authentication or Windows or SQL Server authentication, your SQL Server login credentials might be a Windows account or a SQL Server login account that is not associated with a Windows account.

To facilitate communication and the management of SQL Server logins, you can create Active Directory security groups for the following users and computers:

- **Centrify-Admins** for the user accounts that perform administrative tasks using Audit Manager.
- **Centrify-Auditors** for the user accounts that use Audit Analyzer.
- **Centrify-TrustedCollectors** for the computers accounts that host the collector service.

If you create these Active Directory security groups, you can then use Audit Manager to grant Manage SQL Login permissions for each group to allow its members to connect to the appropriate SQL Server database. Creating Active Directory security groups with SQL Server logins enables you to manage access to the databases required for auditing through Active Directory group membership without the help of the database administrator.

Any time you want to add an administrator, auditor, or collector computer to the installation, you simply add that user account or computer object to the appropriate Active Directory group. If an administrator or auditor leaves or if you want to stop using the collector on a particular computer, you can remove that user or computer from its Active Directory security group to prevent it from accessing the database.

Auditing Security Groups

Centrify-Admins for the user accounts that perform administrative tasks using Audit Manager.	Active Directory	no explicit SQL Server permissions needed — Audit Manager handles the SQL Server permissions	Creating Active Directory security groups with SQL Server logins enables you to manage access to the databases required for auditing through Active Directory group membership without the help of the database administrator.
Centrify-Auditors for the user accounts that use Audit Analyzer.			
Centrify-Collectors for the			

computer accounts that host the collector service.

Determining Storage Requirements for Auditing

There are two important policy decisions your organization must make to determine how much disk space you need for storing audit data and how frequently you should plan to rotate the active database. Early on in the deployment, your organization should consider the following policy decisions:

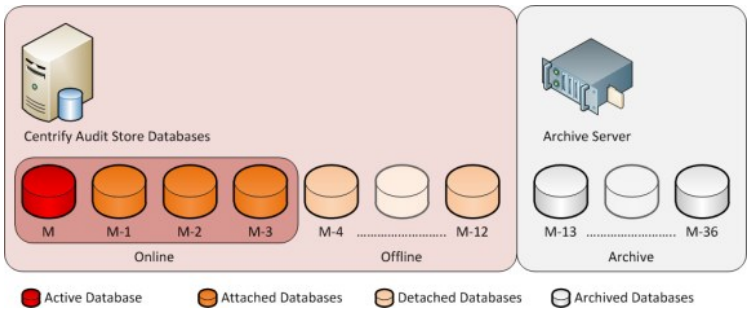
- What is your rotation policy?

  To answer this question, you should decide the period of time audited sessions should be available in the active and attached database for auditors to review using the Audit Analyzer console. For example, you might decide that you want to be able to query audited activity for a minimum of 90 days. Alternatively, you might want to define a rotation policy that is based on the size of the database, so that the active database is not allowed to exceed a specific size. For example, you might decide that the database should not exceed 4GB to optimize performance for archiving.

- What is your retention policy?

  To answer this question, you should decide the period of time to keep audited data available in attached databases and the maximum period of time to keep archived audit data available before purging data that's no longer needed.

To illustrate how these policies affect database management, consider a rotation policy based on a monthly schedule. In this example, an organization decides that audit data must be available for querying for a minimum of 90 days. On the first of each month, a new active database is brought online and the previous 3 months remain available as attached databases to support querying 90 to 120 days of audit data.



In this model, there are four databases online at the same time. This example organization has also decided on a two-stage retention policy. In the first stage, older databases are detached from Audit Analyzer, but remain stored on the SQL Server instance for up to one year. The detached databases provide up to a year of audit history and can be reattached, if that data is needed. In the second stage of the retention policy, the organization archives the audit store databases for up to 3 years. After three years, the oldest data is permanently purged.

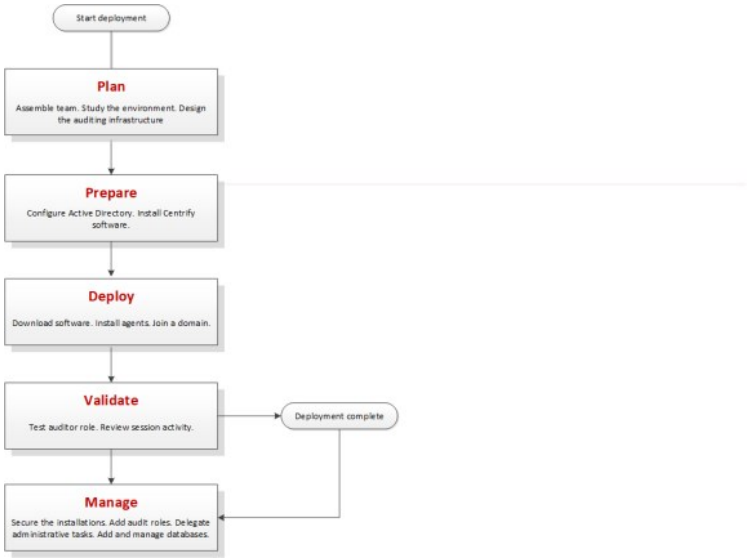
Depending on your requirements, you might use a similar retention policy or have different policies based on the session activity you are capturing. For example, you might keep sessions that capture normal user activity for three years, but keep sessions that capture SOX compliance for ten years.

To project your storage requirements, you will need additional information that is specific to your organization, including the number of computers you plan to audit, the number of sessions that are active on audited computers, and whether you record all activity using video capture or only summaries of user activity. To collect this information, you should monitor a pilot deployment. You can then use the information from the pilot deployment as described in Estimating database requirements based on the data you collect to estimate your storage requirements based on how much audit data you are generating. The decisions you make for the rotation and retention policies will help you further refine those estimations as you expand the deployment.

> **Note**: If you define a rotation policy similar to this example, you can automate the monthly database rotation using Centrify application programming interfaces or using scheduled SQL Server jobs or scripts that perform database maintenance operations. For more information, see the Database Management Guide.

What's Involved in the Deployment Process

Most of the planning in this chapter has focused on designing the auditing infrastructure and deciding where to install components. The following illustration provides a visual summary of the complete deployment process and highlights the keys to success. The sections after the flowchart provide additional details about what's involved in each phase or the decisions you will need to make, such as who should be part of the deployment team, where to install the software, and who has permission to do what.



Plan

During the first phase of the deployment, you collect and analyze details about your organization's requirements and goals. You can then also make preliminary decisions about sizing, network communication, and where to install components.

Here are the key steps involved:

- Identify the goals of the deployment.
  - Is auditing important for specific computers?
  - Is auditing important for computers used to perform administrative tasks?
  - Is auditing important for computers that host specific applications or sensitive information?
  - Should auditing be required for users in specific groups or with specific roles?
- Assemble a deployment team with Active Directory, UNIX, and other expertise, including at least one Microsoft SQL Server database administrator.
- Provide basic training on Centrify architecture, concepts, and terminology.
- Analyze the existing environment to identify target computers where you plan to install Centrify auditing infrastructure components.
  - Plan for permissions and the appropriate separation of duties for your organization.
  - Review network connections, port requirements, firewall configuration.
  - Identify computers for Audit Manager and Audit Analyzer consoles.
  - Identify computers to be used as collectors, audit stores, and the management Database.
  - Verify that you have reliable, high-speed network connections between components that collect and transfer audit data and sufficient disk storage for the first audit store database.
  - Identify the initial target group of computers to be audited.
- Define and document your data archiving and data retention policies.

Prepare

After you have analyzed the environment, you should prepare the Active Directory groups to use. You can then install administrative consoles and the auditing infrastructure.

Here are the key steps involved:

- (Optional) Create the additional Active Directory security groups for your organization.

- Groups can simplify permission management and the separation-of-duties security model.
- Install Audit Manager and Audit Analyzer on at least one administrative Windows computer.
- Create a new audit installation and a management database on one computer.
- Create an audit store and audit store database on at least one computer.
- Install a collector on at least two computers.

Deploy

After you have prepared Active Directory, installed administrative consoles on at least one computer, and created at least one installation, you are ready to deploy agents on the computers to be audited.

Here are the key steps involved:

- Install the agent on the computers you want to audit.
- Join the appropriate domains and zones.
- Prepare a Group Policy Object for deploying agents remotely using a group policy.
- Assign the appropriate permissions to the users and groups who should have access to audit data.

Validate

After you have deployed agents on target computers, you should test and verify operations before deploying on additional computers.

Here are the key steps involved:

- Log on locally to a target computer using an Active Directory user account and password to verify Active Directory authentication.
- Open Audit Analyzer and query for your user session.

Manage

After you have tested and verified auditing operations, you are ready to begin managing your audit installation.

Here are the key steps involved:

- Secure the installation.
- Add auditor roles and assign permissions to the appropriate users and groups.
- Create new databases and rotate the active database.
- Archive and delete old audit data.

Installing the Audit and Monitoring Service

This chapter describes how to install Delinea Audit & Monitoring Service in a production environment. In production environments, you should use a different computer for each component. For example, you should install the collector on its own computer separate from the computer used for the audit store database, and on a separate computer from the audit management database.

To create a simpler installation with all components on the same computer for evaluation purposes, see the Evaluation Guide for Linux and UNIX. For evaluation of auditing features in a Windows-only environment, see the Evaluation Guide for Windows.

Delinea

Installation Preview

As a preview of what's involved in the installation process, the following steps summarize what you need to do and the information you should have on hand for a successful deployment of Centrify software.

To prepare for deployment:

1. Analyze your network topology to determine where to install components and services and any hardware or software updates required.

   For a review of the decisions to make and recommended hardware configuration, see Planning an audit installation.

2. Create a list of the computers where you plan to install different components.

   For example, list the computers where you plan to install agents, collectors, audit store databases, and consoles.

   For a review of the requirements associated with each component, see Planning an audit installation.

3. Determine the scope of the audit installation.

   The most common deployment scenario is a single installation for an Active Directory site, but you can have more than one installation, if needed, and use subnets to limit the scope of the installation.

   For a review of what constitutes an installation, see Deploying auditing components in an audit installation and Deciding on the scope of the installation.

4. Create Active Directory security groups for managing the permissions that are required for accessing the databases that store audit-related information.

   For a review of the Active Directory security groups to create, see Checking SQL Server logins for auditing.

5. Install Microsoft SQL Server.

   If you are not a database administrator in your organization, you should submit a service request or contact an administrator who has permission to create databases.

   For more information about preparing a SQL Server database engine for auditing, see Installing and configuring Microsoft SQL Server for auditing.

6. Install the Audit Manager and Audit Analyzer consoles.

   For more information about installing the consoles, see Installing the Audit Manager and Audit Analyzer consoles.

7. Create a service account with the permissions to create a new installation. For details, see Creating a setup user account for installation.

8. Open Audit Manager to create a new installation.

   For more information about using Audit Manager to create a new installation and audit store, see Creating a new installation.

9. Install the audit collector service on at least two Windows computers.

   You can add collectors to the installation at any time. For more information about installing and configuring collectors, see Installing the audit collectors.

10. Install the Audit Management Server on a Windows computer.

    For more information, see Installing the Audit Management Server.

11. Install a Centrify Agent on each Windows, Linux, or UNIX computer you want to audit.

    For more information about installing Centrify Agents, see Installing the Centrify Agent for Windows and Installing an Centrify Agent for *NIX.

12. Make sure agents are enabled for auditing. For details, see Enabling or disabling auditing on Windows computers and Enabling or disabling auditing on Linux and UNIX computers.

13. Install additional Audit Manager or Audit Analyzer consoles on any Windows computer that you want to use to manage the installation or query and play back session data.

After the initial deployment, you can add new agents, collectors, audit stores, and audit store databases to the installation or create additional installations.

Delinea

Installing and configuring Microsoft SQL Server for Auditing

If you want to audit user activity on Windows, you must have at least one Microsoft SQL Server database instance for the audit management database and audit store databases. Centrify recommends that you use a dedicated instance of SQL Server for the audit management database. A dedicated SQL Server instance is an instance that does not share resources with other applications. The audit store databases can use the same dedicated instance of SQL Server or their own dedicated instances.

There are three database deployment scenarios for your audit installation:

- **Evaluation**—You can install Microsoft SQL Server Express with Advanced features directly from the configuration wizard or by running the SQLEXPRADV_x64_ENU.exe setup program to create a new Microsoft SQL Server Express database instance for testing. However, if you are auditing a production environment*,* you should not use Microsoft SQL Server Express.

  If you choose to install a different version of Microsoft SQL Server Express for an evaluation and the version requires .NET version 3.5 SP1, you will need to manually install the .NET files yourself (the installer doesn't include these files)..

- **Manual installation with system administrator privileges**—Install a Microsoft SQL Server database instance for which you are a system administrator or have been added to the system administrator role.

- **Manual installation without system administrator privileges**—Have the database administrator (DBA) install an instance of Microsoft SQL Server and provide you with system administrator credentials or information about the database instance so that you can create the management database and audit store databases.

Downloading and installing SQL Server manually

You can use an existing instance of Microsoft SQL Server or install a new instance. You can install Microsoft SQL Server directly from the Centrify ISO or ZIP, or download it from the Microsoft web site. In selecting a version of SQL Server to download, you should be sure it includes Advanced Services. Advanced Services are required to support querying using SQL Server full-text search.

After downloading an appropriate software package, run the setup program using your Active Directory domain account and follow the instructions displayed to complete the installation of the Microsoft SQL Server instance.

When selecting the components to install in the setup program, expand the Database Services and select Full Text Search as a feature to be installed. For the authentication mode, select Windows authentication if all connections between auditing components will be in the same forest. If any communication will be outside of the forest, use Mixed Mode authentication and select the option to add the current user to the SQL Server Administrator role.

> **Note**: Centrify does not recommend running SQL Server under a high privilege account such as a LocalSystem account.

Configuring SQL Server to prepare for auditing

After you install the SQL Server database engine and management tools, you should configure the SQL Server instance for auditing. For example, depending on the version of SQL Server you install, you might need to manually enable fulltext search.

To prepare a Microsoft SQL Server database instance for storing audit data:

- Use SQL Server Surface Area Configuration for Services and Connections to check the status and start the database engine, full-text search, and SQL Server Browser services.
- Use SQL Server Surface Area Configuration for Services and Connections or SQL Server Configuration Manager to enable remote connections for TCP/IP.
- Verify whether SQL Native Client Configuration Client Protocol is using the default TCP port 1433 for network communications. If you use a different port, you should note the port number because you will need to specify it in the server name when you create the management and audit store databases.
- Use SQL Server Configuration Manager to restart the SQL Server and SQL Server Browser services.
- Create a database backend service account in the system administrator (sa) fixed server role on the selected database server; you'll specify this account when you create the audit installation. This account is used to run backend stored procedures. If desired, this can be the same account that you use the create the audit installation, as mentioned in Creating a setup user account for installation.

Configuring Amazon RDS for SQL Server for auditing

You can deploy audit store databases on Amazon RDS instances, if desired. Centrify supports Amazon RDS for 2016 and earlier versions (not 2017).

You must host the audit management database on a traditional SQL Server, such as SQL Server Express, Standard, or Enterprise.

If you want to use an instance of Amazon RDS for SQL Server for audit store databases you need to do the following configurations:

- After you set up your Amazon RDS for SQL Server, join the RDS SQL server to AWS Microsoft Active Directory.
- Enable these DB Parameter Group settings on RDS SQL Server:
  - clr enabled
  - show advanced options

    You can use the AWS Management Console, API, or the AWS command line interface to enable these settings.

    For more details, see <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html>.
- Set up a one-way or two-way forest trust between the AWS Microsoft Active Directory and your on-premise Active Directory forest so that users of your on-premise Active Directory forest can access resources in the AWS Microsoft Active Directory.

  Note: Amazon RDS for SQL Server with High Availability is supported.

Amazon RDS for SQL Server required permissions

The permissions for Amazon RDS for SQL Server vary a little from the permissions for local or network instances of SQL Server. This section covers the Amazon RDS for SQL Server permission required or granted for each auditing component.

Permissions to the audit store database stored procedures service account

The stored procedures service account (in other words, the 'execute as' account) no longer requires the sysadmin server role permission if the audit store database is on Amazon RDS for SQL Server.

The service account requires only the db_owner database role permission and the account will be added to be member of db_owner database role by Add Audit Store Database wizard.

> **Note**:You do not need to grant the permissions manually. The Audit Manager console, Powershell cmdlet, or SDK grants the permissions to the service account.

Collector account permissions for audit store databases on Amazon RDS for SQL Server

The collector account requires the following server level permissions on the Amazon RDS for SQL Server:

- 'View Any Definition' server level permission
- 'View Server State' server level permission

The collector account requires the following database level permissions on the audit store database:

- A member of the 'collector' database role

  > **Note**:You do not need to grant the permissions manually. The Audit Manager console, Powershell cmdlet, SDK, or the Collector Configuration wizard grants the permissions to the collector account.

Management Database Account permissions for audit store databases on Amazon RDS for SQL Server

The management database account requires the following server level permissions on the RDS SQL server:

- 'Alter Trace' server level permission
- 'Alter Any Login' server level permission
- Grant permission of 'Alter Any Login' server level permission
- Grant permission of 'View Any Definition' server level permission
- Grant permission of 'View Server State' server level permission

The management database account requires the following database level permissions on the audit store database:

- A member of 'managementdb' database role

Note:You do not need to grant the permissions manually. The Audit Manager console, Powershell cmdlet, or SDK grants the permissions to the management database account.

Permissions to create the audit store database on Amazon RDS for SQL Server

In order to create an audit store database on Amazon RDS for SQL Server, you must have the following permissions:

- 'Create Any Database' server level permission to create the database on the server
- 'Alter Any Login' server level permission to create the login for the management database account and the collector account
- Alter Any Login' server level permission to grant the 'Alter Any Login' permission to the management database account
- 'Alter Trace' server level permission to grant the 'Alter Trace' permission to the management database account
- 'View Any Definition' server level permission to grant the 'View Any Definition' (with grant) permission to the management database account and also to grant the 'View Any Definition' permission to the collector account
- Grant permission of 'View Server State' server level permission to grant the 'View Server State' (with grant) permission to the management database account and also to grant the 'View Server State' permission to the collector account

Permissions to upgrade the audit store database on Amazon RDS for SQL Server

The required permission to upgrade the audit store database on Amazon RDS for SQL Server is the 'db owner' permission on the database. No server level permissions are required

Installing the Audit Manager and Audit Analyzer Consoles

You can install Audit Manager and Audit Analyzer on the same computer or on different computers. The computers where you install the consoles must be joined to the Active Directory domain and be able to access the management database.

In most cases, the consoles are installed together on at least one computer.

You can use either the individual console installers or the main installer.

- Install Audit Manager using the console installer
- Install Audit Analyzer using the console installer
- Install both consoles using the main installer

Install Audit Manager Using the Individual Console Installer

1. Log on using an Active Directory domain account.
2. Open the ISO file, and navigate to the following folder:

   \DirectAudit\Console\
3. Run the Audit Manager installer : Centrify DirectAudit Administrator Console64.exe.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
6. On the Destination Folder page, review the installation location and click **Next** to continue.
7. If you want to install the console in a different location, click **Change** and navigate to the desired folder.
8. Click **Install** to begin the installation.
9. The installer installs the necessary files. To open the console, keep the **Launch Centrify Audit Manager** option selected. Otherwise, deselect the option.
10. Click **Finish** to close the installer.

After you install Audit Manager, you can open Audit Manager to create a new installation.

Install Audit Analyzer Using the Individual Console Installer

1. Log on using an Active Directory domain account.
2. Open the ISO file, and navigate to the following folder:

   \DirectAudit\Console\
3. Run the Audit Manager installer : Centrify DirectAudit Auditor Console64.exe.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
6. On the Destination Folder page, review the installation location and click **Next** to continue.
7. If you want to install the console in a different location, click **Change** and navigate to the desired folder.
8. Click **Install** to begin the installation.
9. The installer installs the necessary file. To open the console, keep the **Launch Centrify Audit Analyzer** option selected. Otherwise, deselect the option.
10. Click **Finish** to close the installer.

Install Audit Manager and Audit Analyzer on the Same Computer Using the Main Installer

1. Log on using an Active Directory domain account.
2. Open the ISO file.

   If you created a physical CD from the ISO file that you downloaded, the Getting Started page is displayed automatically. If the page is not displayed, open the autorun.exe file to start the installation.
3. On the Getting Started page, click **Audit & Monitor** to start the setup program for audit and monitoring service components.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
6. Select **Centrify Administration** to install both Audit Manager and Audit Analyzer, then click **Next**.
7. In the rare case where the administrator should not have access to the Audit Analyzer, select Audit Manager, then click **Next**.
8. After you install Audit Manager, you are prompted to create a new installation. If you want to create the installation at a later time, you can run the setup program again to create a new installation.

Creating a Setup User Account for Installation

You'll need to create an account to use when you create the audit installation, set up audit stores, and so forth. This account needs to have the following permissions:

- Active Directory:
  - Permission to create serviceConnectionPoint objects on the container or organizational unit you select for publishing installation information
- SQL Server:
  - Be a member of the system administrator (sa) fixed server role

This user account needs these permissions for the initial installation and some maintenance tasks. It's a good practice to add this account to your Centrify-admins security group, as mentioned in Creating security groups for auditing.

Creating a New Installation

Before you can begin auditing, you must create at least one audit installation and a management database. Creating the management database, however, requires SQL Server system administrator privileges on the computer that hosts the SQL Server instance. If possible, you should have a database administrator add your Active Directory domain account to the SQL Server system administrators role.

If you have not been added to the system administrators role, you should contact a database administrator to assist you. For more information about creating a new installation when you don't have system administrator privileges, see How to create an installation without system administrator privileges.

To Create a New Installation and Management Database as a System Administrator

1. Log on using an Active Directory account with permission to install software on the local computer and permissions listed in Creating a setup user account for installation.
2. Open Audit Manager.

   Note:If you haven't configured an audit installation yet, the New Installation wizard opens automatically.
3. If this isn't your first audit installation: in Audit Manager, right-click **Centrify Audit Manager** and select **New Installation** to open the New Installation wizard.
4. Enter a name for the new installation, then click **Next**.

   Tip: Name the installation to reflect its administrative scope. For example, if you are using one installation for your entire organization, you might include the organization name and All or Global in the installation name, such as AcmeAll. If you plan to use separate installations for different regions or divisions, you might include that information in the name, for example AcmeBrazil for a regional installation or AcmeFinance for an installation that audits computers in the Finance department.
5. Select the option to create a new management database and verify the SQL Server computer name, instance name, and database name are correct.

   If the server does not use the default TCP port (1433), you must provide the server and instance names separated by a backslash, then type a comma and the appropriate port number. For example, if the server name is ACME, the instance name is BOSTON, and the port number is 1234, the server name would be ACME\BOSTON,1234.

   If you're installing on a SQL cluster, enter the SQL cluster name in the SQL Server computer name field.

   If you're connecting to a SQL Server availability group listener, click Options (next to the Server Name) and enter the following connection string parameters:

   MultiSubnetFailover=Yes

   Click **Next** to continue.
6. Select **Use the default NT AUTHORITY\SYSTEM account** to use the internal account or select a specific SQL login account with sufficient privileges, then click **Next**.

   A SQL login account is required to run the stored procedures that read and write information to the management database. The account must a member of the system administrator (sa) fixed server role on the selected database server, as mentioned in Configuring SQL Server to prepare for auditing.
7. Type the license key you received, then click **Add** or click **Import** to import the keys directly from a file, then click **Next**.
8. Accept the default location or click **Browse** to select a different Active Directory location for publishing installation information, then click **Next**.

   You must have the Active Directory permission to Create serviceConnectionPoint objects on the container or organizational unit you select for publishing installation information.
9. Select the installation-wide auditing options you want to enable, then click **Next**.
   - Select **Enable video capture recording of user activity** if you want to capture shell or desktop activity on computers when users are audited, then click **Next**.

   Selecting this option enables you to review everything displayed during an audited user session, but will increase the audit store database storage

requirements for the installation. You can deselect this option if you are only interested in a summary of user activity in the form of audit trail events. Audit trail events are recorded when users log on, open applications, and select and use role assignments with elevated rights.

- Select **Do not allow any users to review their own sessions** to prevent all users from updating the review status for their own sessions or adding comments to their own sessions.
- Select **Do not allow any users to delete their own sessions** to prevent all users from deleting their own sessions.

If you set either of the installationwide policies disallowing user activity, the policy takes precedence over any rights provided by a user's audit role.

10. Review details about the installation and management database, then click **Next**.

If you have SQL Server system administrator (sa) privileges and can connect to the SQL Server instance, the wizard automatically creates the management database.

11. Select the **Launch Add Audit Store Wizard** option if you want to start the Add Audit Store wizard, then click **Finish**

If you want to create the first audit store database on a different SQL Server instance, you should deselect the **Launch Add Audit Store Wizard** option and click **Finish**.

For more information about adding the first audit store database, see Creating the first audit store.

How to Create an Installation without System Administrator Privileges

If you do not have the appropriate permission to create SQL Server databases, you cannot use the New Installation wizard to create the management database without the assistance of a database administrator.

If you do not have system administrator privileges, the wizard prompts you to specify another set of credentials or generate SQL scripts to give to a database administrator. For example:

If you don't have a database administrator immediately available who can enter the credentials for you, you cannot continue with the installation.

To Create an Installation when you don't have System Administrator Privileges

1. Select the option to generate the SQL scripts, then click **Next**.
2. Select the folder location for the scripts, then click **Next**.
3. Review details about the installation and management database you want created, then click **Next**.

   The wizard generates two scripts: Script1 prepares the SQL Server instance for the management database and Script2 creates the database.
4. Click **Finish** to exit the New Installation wizard.
5. Send the scripts to a database administrator with a service or change-control request.

   Note:You should notify the database administrator that the scripts must be run in the proper sequence and not modified in any way. Changes to the scripts could render the database unusable.
6. After the database administrator creates the database using the scripts, open the Audit Manager console to run the New Installation wizard again.
7. Type the name of the installation, then click **Next**.
8. Select **Use an existing database** and verify the database server and instance name, then click the Database name list to browse for the database

name that the database administrator created for you.

If the server does not use the default TCP port, specify the port number as part of the server name. For example, if the port number is 1234, the server name would be similar to ACME\BOSTON,1234.

If you're installing on a SQL cluster, enter the SQL cluster name in the SQL Server computer name field.

9. Select the database name from the list of available databases, click **OK**, then click **Next**.

   You should only select an existing database if the database was created using scripts provided by Centrify.

10. Select **Use the default NT AUTHORITY\SYSTEM account** to use the internal account or select a specific SQL login account with sufficient privileges, then click **Next**.

A SQL login account is required to run the stored procedures that read and write information to the management database. The account must a member of the system administrator (sa) fixed server role on the selected database server.

11. Type a license key or import licenses from a file, then click **Next**.
12. Review details about the management database to be installed, then click **Next**.
13. Select the **Launch Add Audit Store Wizard** option if you want to start the Add Audit Store wizard, then click **Finish**.

Creating the First Audit Store

If you selected the Launch Add Audit Store Wizard check box at the end of the New Installation Wizard, the Add Audit Store Wizard opens automatically. You can also open the wizard at any time by right-clicking the Audit Stores node in the Audit Manager console and choosing Add Audit Store.

To create the first audit store:

1. Type a display name for the audit store, then click **Next**.

   Tip: If your plan specifies multiple audit stores, use the name to reflect the sites or subnets serviced by this audit store. Note that an audit store is actually a record in the management database. It is not a separate process running on any computer. You use a separate wizard to create the databases for an audit store.

2. Select the type of systems that the audit store will serve.

   You can choose to separate Windows traffic from UNIX traffic if both types of agents belong to the same site or subnet.

   The options are:

   - Windows and UNIX
   - Windows
   - UNIX

   Click **Next** to continue.

3. Click **Add Site** or **Add Subnet** to specify the sites or subnets in this audit store.

   - If you select Add Site, you are prompted to select an Active Directory site.
   - If you select Add Subnet, you are prompted to type the network address and subnet mask.

   After you make a selection or type the address, click **OK**. You can then add more sites or subnets to the audit store. When you are finished adding sites or subnets, click **Next** to continue.

   The computer you use to host the audit store database should be no more than one gateway or router away from the computers being audited. If your Active Directory sites are too broad, you can use standard network subnets to limit the scope of the audit store.

4. Review information about the audit store display name and sites or subnets, then click **Next**.

5. Select the **Launch Add Audit Store Database Wizard** option if you want to create the first audit store database, then click **Finish**.

Creating the First Audit Store Database

If you selected the Launch Add Audit Store Database Wizard check box at the end of the Launch Add Audit Store Wizard, the Add Audit Store Database Wizard

opens automatically. You can also open the wizard at any time from the Audit Manager console by expanding an audit store, right-clicking the Databases node, and choosing Add Audit Store Database.

To create the first audit store database:

1. Type a display name for the audit store database, then click **Next**.

   The default name is based on the name of the audit store and the date the database is created.

2. Select the option to create a new database and verify that the SQL Server computer name, instance name, and database name are correct.

   The default database name is the same as the display name. You can change the database name to be different from the display name, if you want to use another name.

   If the server does not use the default TCP port, specify the port number as part of the server name. For example, if the port number is 1234, the server name would be similar to ACME\BOSTON,1234.

   If you're installing on a SQL cluster, enter the SQL cluster name in the SQL Server computer name field.

   When entering the SQL Server host computer name, note that you can enter either the server short name (which is automatically resolved to its fully qualified domain name, or FQDN) or the actual server FQDN or the CNAME alias for the server.

   If the database is an Amazon RDS SQL Server:

   1. Select the **This is an Amazon RDS SQL Server** option.
   2. In the Server Name field, enter the RDS SQL Server database instance endpoint name used for Kerberos authentication.

   For example, if the database host name is northwest1 and the domain name is sales.acme.com, then the endpoint name would be northwest1.sales.acme.com.

   Click **Options** to enter additional connection string parameters or to enable data integrity checking.

   - If you're connecting to a SQL Server availability group listener, click Options (next to the Server Name) and enter the following connection string parameters:

   MultiSubnetFailover=Yes

   - You can enable or disable data integrity checking once, when you create the audit store database. To change the state, you must rotate to a new audit store database.

   When you create your audit store database, you have the option to enable data integrity checking. Data integrity checking provides the ability to detect if auditing data has been tampered. For example, data integrity checking can detect if a user who has write privileges over the Audit Store database directly manipulates the audited session data by making a direct connection to the Microsoft SQL Server database. Data integrity checking cannot detect tampering if a database administrator deletes an entire session or database.

   Click **Next** to continue.

3. Because this is the first audit store database, you also want to make it the active database. This option is selected by default. If you are creating the database for future use and don't want to use it immediately, you can deselect the **Set as active database** option. The option to create a new database is also selected by default.

   Click **Next** to continue.

4. Specify the stored procedures services account:

   - Select **Use the default NT AUTHORITY\SYSTEM account** to use the internal account
   - Or, select **Specify a SQL Login account** and enter a specific SQL login account with sufficient privileges.

   A SQL Server login account is required to run the stored procedures that read and write information to the management database.

   For local or network databases, the account must a member of the system administrator (sa) fixed server role on the selected database server.

   If the database is an Amazon RDS for SQL Server, the account you specify will be added as a member of the db_owner fixed database role in Amazon RDS for SQL Server.

Click **Next** to continue.

5. Review details about the audit store database, then click **Next**.

   If you have the correct privileges and can connect to the SQL Server instance, the wizard automatically creates the audit store database.

Installing the Audit Collectors

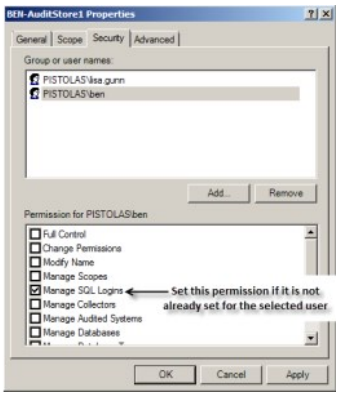
After you have created a new installation, with an audit management database and at least one audit store and audit store database, you must add the collectors that will receive audit records from the agents and forward those records to the audit store. For redundancy and scalability, you should have at least two collectors. For more information about planning how many collectors to use and the recommended hardware and network configuration for the collector computers, see Deciding where to install collectors and audit stores.

Set the Required Permission

Before you configure a collector, you should check whether your user account has sufficient permissions to add new collector accounts to the audit store database. If you are a database administrator or logged on with an account that has system administrator privileges, you should be able to configure the collector without modifying your account permissions. If you have administrative rights on the computer that hosts Audit Manager but are not a database administrator, you can set the appropriate permission before continuing.

To set the permission required to add accounts to the audit store database:

1. Open Audit Manager.
2. Expand the installation, then expand Audit Stores.
3. Select the audit store that the collector will connect to, right-click, then click **Properties**.
4. Click the **Security** tab.
5. Click **Add** to search for and select the user who will configure the collector.
6. Select the **Manage SQL Logins** right, then click **OK**.



Install the Collector Service using the Setup Program

If your user account has sufficient permissions to add new collector accounts to the audit store database, you can install a collector by running the setup program on the computer on which you want to install the collector. When you are prompted to select components, select Audit Collector and deselect all of the other components, then click **Next**. Follow the instructions in the wizard to select the location for installing files and to confirm your selections, then click **Finish** to complete the installation.

The collector installer is in the \DirectAudit\Collector folder in your installation media.

Configure the Audit Collector Service

By default, when you click **Finish**, the setup program opens the Collector Configuration Wizard. Alternatively, you can launch the configuration wizard at any time by clicking **Configure** in the Collector Control Panel.

To configure the collector service:

1. On the first screen of the Collector Configuration Wizard, select the DirectAuditinstallation to assign this collector to.

   If the computer is also enrolled in the Centrify Cloud Platform and you have already enabled auditing in the Admin Portal, you can choose which kind of

audit installation to assign the collector to:

- **Automatic**: This option configures the collector to receive audit data from systems that are enrolled in the Centrify Cloud Platform and systems that are joined to Active Directory.

  You use the Admin Portal to configure which installation is used by these systems. The systems have either the Centrify Client for Linux or Centrify Client for Windows and the audit packages installed so that auditing is enabled. These systems do not have to be joined to Active Directory.

- **Manual**: This option configures the collector to receive audit data from systems that are joined to Active Directory and have either the Centrify Agent for *NIX or Centrify Agent for Windows installed and the system is enabled for auditing. For this option, select the audit installation.

  Computers that are not enrolled in the Centrify Cloud Platform have a single list of audit installations to pick from.

Click **Next** to continue.

The configuration wizard verifies that the specified installation has an audit store that services the site that the collector is in and that the collector and its audit store database are compatible.

2. Enter the port number(s) that the collector will use to communicate with the audited systems.
   - The default port is 5063 for systems that have either the Centrify Agent for *NIX or Centrify Agent for Windows installed.
   - If the computer is also enrolled in the Centrify Cloud Platform, the default port is 5064 for systems that have either the Centrify Client for Linux or Centrify Client for Windows installed.
   - If you set the installation to Manual in the previous step, Centrify Client System port is greyed out.

     For either port, if you specify a different port and have the default Windows firewall turned on, the wizard checks whether the port is open. If the port isn't open, the wizard offers to open it for you.

     If you are using another vendor's firewall, open the port with the tools provided by that vendor. If there's an upstream firewall—such as a dedicated firewall appliance—between the collector and the computers to be audited, contact the appropriate personnel to open the port on that firewall.

   Click **Next** to continue.
3. If the computer where you're configuring a collector belongs to multiple audit stores in the auditing installation, choose which audit store this collector will connect to, then click **Next**.

   For example, two audit stores can have an overlapping scope if one audit store scope is configured for Active Directory sites and another one is set by subnets.
4. Select whether you want to use Windows authentication or SQL Server authentication when the collector authenticates to the audit store database, then click **Next**.

   In most cases, you should choose Windows authentication to add the computer account to the audit store database as a trusted, incoming user.

   If Microsoft SQL Server is in a different forest or in an untrusted forest, you should use SQL Server Management Studio to set up one or more SQL Server login accounts for the collector. After you create the SQL Server login account for the collector to use, you can select SQL Server authentication, then type the SQL Server login name and password in the wizard.
5. Type the maximum number of connections for the Microsoft SQL Server connection pool, then click **Next**.
6. Review the settings for the collector, then click **Next**.
7. Click **Finish** to close the wizard and start the collector service.

Installing the Audit Management Server

It's a best practice to install the Audit Management Server after you've installed your audit stores, audit store databases, and installed your collectors. You can install the Audit Management Server either on a new computer or one where you've installed a collector.

To install the Audit Manager Server:

1. Log on using an Active Directory domain account.
2. Open the ISO file, and navigate to the following folder:

   \DirectAudit\Audit Management Server\
3. Run the Audit Management Server installer : Centrify DirectAudit Audit Management Server64.exe.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
6. On the Destination Folder page, review the installation location and click **Next** to continue.
7. If you want to install the console in a different location, click **Change** and navigate to the desired folder.
8. Click **Install** to begin the installation.
9. The installer installs the necessary files. To open the console, keep the **Run Audit Management Configuration Wizard** option selected. Otherwise, deselect the option.
10. Click **Finish** to close the installer.

Configuring the Audit Management Server

By default, when you finish installing the Audit Management Server, the installer opens the Audit Management Server Configuration Wizard.

To configure the Audit Management Server:

1. On the first page, select the installation for which you want to configure the Audit Management Server.

   If you have one installation, it's already selected.

   Click **Next** to continue.
2. On the Authentication Type page, specify which kind of authentication that the Audit Management Server will use. The choices are:

   - **Windows Authentication:** Specify the computer account that will run the Audit Management Server.
   - **SQL Server Authentication**: Specify the SQL Server user name and password to use. Click Test Connection to make sure that the login credentials work.

   Click **Next** to continue.
3. On the Summary page, review the Audit Management Server configuration details, and click **Next** to continue.
4. Click **Finish** to close the configuration wizard.

Installing the Centrify Agent for Windows

You must install an agent on every Windows computer that you want audit. You can install the agent in the following ways:

- **Interactively**, by running the Centrify setup program on each computer.

  When the installation finishes, the agent configuration wizard launches automatically. You can configure the agent right away, or exit the configuration wizard and configure the agent later. See Installing interactively using the setup program for details.

- **Silently**, by executing appropriate commands in a terminal window on each computer.

  You can installation silently on a local computer or use a software distribution product, such as Microsoft System Center Configuration Manager (SCCM), to execute the appropriate commands remotely to deploy agents on remote computers. After installation, you can change the default agent settings, if needed. See Installing silently by using the Microsoft Windows Installer for details.

- **Silently and centrally**, by using a group policy to execute commands remotely on the computers in a domain or organizational unit.

  If you use the Centrify Group Policy Deployment files, you can both install and configure the registry on remote computers from a central location without a separate software distribution product. However, you must configure the Windows agent registry settings in a file before deploying. See Installing from a central location by using group policy for details.

Regardless of the deployment method you choose, you should first make sure that the computers where you plan to deploy meet all of the installation prerequisites.

Delinea

Verify Prerequisites

Before installing the Windows agent, verify the computer on which you plan to install meets the following requirements:

- The computer is running a supported Windows operating system version.
- The computer is joined to Active Directory.
- The computer has sufficient processing power, memory, and disk space for the agent to use.
- The computer has the .NET Framework, version 4.5.2 or later.
- The computer has Windows Installer version 3.1, or later.

If you are installing interactively using the setup program, the setup program can check that the local computer meets these requirements and install any missing software. If you are installing silently from the command line or by using a Group Policy Object, you should verify the computers where you plan to install meet these requirements. If you are installing silently and a computer does not meet these requirements, the installation will fail.

Installing Interactively Using the Setup Program

If you select auditing when you install the Windows agent, the agent starts capturing user session activity immediately after it is installed. Therefore, you should be sure that you have an installation, audit store database, and collector prepared and available before installing an agent. If the agent cannot connect to an installation, it stores the captured session data locally and can quickly overload the local computer's resources.

To install the agent on Windows using the setup program:

1. Log on to the computer and insert the CD or browse to the location where you have saved downloaded Centrify files.

   If the Getting Started page is not displayed automatically, open the autorun.exe file.
2. On the Getting Started page, click **Agent** to start the setup program for the Windows agent.
3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement, click **I accept the terms in the License Agreement**, then click **Next**.
5. Verify the location where files will be installed, then click **Next**.

   If you want to install in a location other than the default location, click Browse, select a different location, then click **Next**.
6. Click **Install**.
7. Click **Finish** to complete the installation and start the agent configuration wizard.
8. In the Centrify Agent Configuration window, click **Add Service**.

9. In the dialog box that opens, select the Centrify Auditing and Monitoring Service option and click **OK**.
10. In the Enable session capture and replay window, select the auditing installation to which you want the agent on this computer to connect.

    Click **Next** to continue.

    The Centrify Auditing and Monitoring Service is now listed as an enabled service.
11. Close the Agent configuration window and click **Exit** in the installer window.

Configuring the Agent Settings for Auditing

The agent configuration wizard automatically configures several default settings in the agent registry. If you want to view or change the agent settings for auditing on a Windows computer after running the configuration wizard—or if you did not use the configuration wizard immediately after installation—you can use the Agent Configuration Wizard.

To configure the agent settings for auditing:

1. Click **Start > All Programs > Centrify Server Suite 2021.1 > Centrify Agent for Windows Configuration > Agent Configuration**.
2. In the Centrify Agent Configuration window, locate the Centrify Auditing and Monitoring Service option, and click **Settings**.

   The Centrify Auditing and Monitoring Service Settings window opens.
3. On the General tab, click **Configure**.
4. Select the maximum color quality for recorded sessions, then click **Next**.

   If your audit installation has video capture auditing enabled, you can configure the color depth of the sessions to control the size of data that must be transferred over the network and stored in the database. A higher color depth increases the CPU overhead on audited computers but improves resolution when the session is played back. A lower color depth decreases network traffic and database storage requirements, but reduces the resolution of recorded sessions.

   The default color quality is Low (8-bit).
5. Specify the offline data location and the maximum percentage of disk that the offline data file should be allowed to occupy, then click **Next**.

If the agent cannot connect to a collector, it saves session activity in the offline data location you specify until it can contact a collector.

The spool threshold defines the minimum percentage of disk space that should be available to continue auditing. It is intended to prevent audited computers from running out of disk space if the agent is sending data to its offline data storage location because no collectors are available.

For example, if you set this threshold to 10%, auditing will continue while spooling data to the offline file location as long as there's at least 10% disk space is available on the spool partition. When the disk space available reaches the threshold, auditing will stop until a collector is available.

The agent checks the spool disk space by periodically running a background process. By default, the background process runs every 15 seconds. Because of the delay between background checks, it is possible for the actual disk space available to fall below the threshold setting. If this were to occur, auditing would stop at the next interval. You can configure the interval for the background process to run by editing the HKLM\Software\Centrify\DirectAudit\Agent\DiskCheckInterval registry setting.

6. Select the installation that the agent belongs to, then click **Next**.
7. If the computer where you're configuring an agent belongs to multiple audit stores in the auditing installation, choose which audit store this agent will connect to, then click **Next**.
8. In the Summary page, review your settings, then click **Next**.

   The agent is now configured and enabled for auditing.
9. Click **Finish** to close the agent configuration wizard, then click **Close** to exit the Centrify Auditing and Monitoring Service Settings window.

Deciding to Install With or Without Joining the Computer to a Zone

Before you begin a silent installation, you should decide whether you will wait until later to join the computer to a zone, or join the computer to a zone as part of the installation procedure.

If you install without joining a zone during installation:

- See Installing silently by using the Microsoft Windows Installer for details about the registry settings that you can configure manually after the installation finishes.
- See Installing silently without joining a zone for details about performing the installation.

If you install and join a zone during installation:

- You use a transform (MST) file that is provided with Server Suite to configure a default set of agent-specific registry keys during the silent installation.
- You can optionally edit the MST file before performing the installation to customize agent-specific registry settings for your environment.
- You can optionally use the agent configuration control panel or the registry editor to configure registry settings after the installation finishes.
- See Installing silently by using the Microsoft Windows Installer for details about the registry settings that you can configure by editing the MST file.
- See Installing silently by using the Microsoft Windows Installer for details about how to edit the MST file before you perform the installation.
- See Installing and joining a zone silently for details about performing the installation.

Installing Silently Without Joining a Zone

This section describes how to install the agent silently without joining the computer to a zone. This procedure includes configuring registry settings manually using the registry editor or a third-party tool.

> **Note**:To install the agent and join the computer to a zone during installation, see Installing and joining a zone silently for more information.

Check prerequisites:

1. Verify that the computers where you plan to install meet the prerequisites described in Verify prerequisites. If prerequisites are not met, the silent installation will fail.
2. If you are installing audit and monitoring service, verify that the following tasks have been completed:
    1. Installed and configured the SQL Server management database and the SQL Server audit store database.
    2. Installed and configured one or more collectors.
    3. Configured and applied the Centrify DirectAudit Settings group policy that specifies the installation name.

To install the Centrify Agent for Windows silently without joining the computer to a zone:

1. Open a Command Prompt window or prepare a software distribution package for deployment on remote computers.

    For information about preparing to deploy software on remote computers, see the documentation for the specific software distribution product you are using. For example, if you are using Microsoft System Center Configuration Manager (SCCM), see the Configuration Manager documentation.

2. Run the installer for the Centrify Agent for Windows package. For example:

    msiexec /qn /i "Centrify Agent for Windows64.msi"

    By default, none of the services are enabled.

3. Use the registry editor or a configuration management product to configure the registry settings for each agent.

    For example, under HKEY_LOCAL_MACHINE\Software\Centrify\DirectAudit\Agent, you could set the DiskCheckThreshold key to a value other than the default value of 10%.

To install the Centrify Agent for Windows and add a computer to a zone during installation:

1. Prepare a computer account in the appropriate zone using Access Manager or the PowerShell command New-CdmManagedComputer.

2. You will use the default transform file Group Policy Deployment.mst in Step 3 to update the MSI installation file so that the computer is joined to the zone in which it was pre-created in Step 1. You can optionally modify Group Policy Deployment.mst to change or add additional registry settings during installation.

    If you want to edit Group Policy Deployment.mst to change or add additional registry settings and have not yet done so, edit it now as described in

Installing silently by using the Microsoft Windows Installer.

In order for the computer to join the zone from Step 1, the Group Policy Deployment.mst file *must* specify the GPDeployment property with a value of 1.

3. Run the following command:

   msiexec /i "Centrify Agent for Windows64.msi" /qn TRANSFORMS="Group Policy Deployment.mst"

Installing and Joining a Zone Silently

This section describes how to install the agent and join the computer to a zone at the same time. The procedure described here includes the following steps in addition to executing the MSI file:

- You first prepare (pre-create) the Windows computer account in the appropriate zone.

You execute an MST file together with the MSI file to join the computer to a zone and configure registry settings during the installation.

Installing silently by Using the Microsoft Windows Installer

If you want to perform a "silent" (also called *unattended*) installation of the Centrify Agent for Windows, you can do so by specifying the appropriate command line options and Microsoft Windows Installer (MSI) file to deploy. You must execute the commands on every Windows computer that you want to audit.

You can also use silent installation commands to automate the installation or upgrade of the Windows agent on remote computers if you use a software distribution product, such as Microsoft System Center Configuration Manager (SCCM), that enables you to run commands remotely to deploy software packages. However, only the command-line instructions are covered in this guide.

Configuring Registry Settings

When you perform a silent installation, several registry settings specific to the agent are configured by the default MSI file. In addition, a default transform (MST) file is provided for you to use if you join the computer to a zone as part of the installation procedure. When executed together, the default MSI and MST files ensure that the computer is joined to a zone, and that a default set of agent-specific registry keys is configured.

If your environment requires different or additional registry settings, you can edit the MST file before performing an installation. Then, when you execute the MSI and MST files to perform an installation, your customized registry settings are implemented. For details about how to edit the MST file, see Editing the default transform (MST) file.

> **Note**:If you do not join the computer to a zone during installation, you do not use the MST file. In this situation, you can create or edit registry keys manually after the installation finishes by using the , or the registry editor.

The following table describes the agent-specific registry settings that are available for you to configure during installation (by using the MST file) or after installation (by using the or the registry editor). Use the information in this table if you need to configure registry settings differently than how they are configured by the default MSI and MST files. Keep the following in mind as you review the information in the table:

- The default MSI file is named Centrify Agent for Windows64.msi, and is located in the **Agent** folder in the Centrify download location.
- The default MST file is named Group Policy Deployment.mst, and is located in the **Agent** folder in the Centrify download location.
- All of the settings in the following table are optional, although some are included in the default MSI and MST files so that they are configured when the MSI and MST files execute during an installation.
- Settings that are included in the default MSI and MST files are noted in the table.
- Some settings are environment-specific, and therefore do not have a default value. Others are not environment-specific, and do have a default value.
- The settings described in the table are located in the MSI file's Property table.
- The **Setting** column shows both the property name in the MSI file, and the name (in parentheses) of the registry key in the Windows registry.

Auditing and Monitoring	REG_MAX_FORMAT (MaxFormat)	Specifies the color depth of sessions recorded by the agent. The color depth affects the resolution of the activity recorded and the size of the records stored in the audit store database when you have video capture auditing enabled. You can set the color depth to one of the following values: 0 to use the native color depth on an audited computer. 1 for a low resolution with an 8-bit color depth 2 for medium resolution with a 16-bit color depth (default) 4 for highest resolution with a 32-bit color **This setting is included in the default MSI file.** In the registry, this setting is specified by a numeral (for

		example, 1). In the MSI file Property table, it is specified by the # character and a numeral (such as #1). The default value is 1.
Auditing and Monitoring	REG_DISK_CHECK_THRESHOLD (DiskCheckThreshold)	Specifies the minimum amount of disk space that must be available on the disk volume that contains the offline data storage file. You can change the percentage required to be available by modifying this registry key value. **This setting is included in the default MSI file.** In the registry, this setting is specified by a numeral (for example, 1). In the MSI file Property table, it is specified by the # character and a numeral (such as #10). The default value is 10, meaning that at least 10% of the disk space on the volume that contains the offline data storage file must be available. If this threshold is reached and there are no collectors available, the agent stops spooling data and audit data is lost.
Auditing and Monitoring	REG_SPOOL_DIR (SpoolDir)	Specifies the offline data storage location. The folder location you specify will be where the agent saves ("spools") data when it cannot connect to a collector. **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that it is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Auditing and Monitoring	REG_INSTALLATION_ID (InstallationId)	Specifies the unique global identifier (GUID) associated with the installation service connection point. **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that it is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Auditing and Monitoring	REG_LOG_LEVEL_DA (LogLevel)	Specifies what level of information, if any, is logged. Possible values are: off information warning error verbose **This setting is included in the default MSI file.** The default value is information.
Authentication & Privilege	REG_RESCUEUSERSIDS (RescueUserSids)	Specifies which users have rescue rights. Type user SID strings in a comma separated list. For example: *user1SID*,*user2SID*,*usernSID* **This setting is not included in the default MSI file.** To use it, you must edit the default transform (MST) file so that the setting is processed together with the MSI file during installation, or create it manually in the registry after the installation finishes.
Authentication & Privilege	REG_LOG_LEVEL_DZ (LoggingLevel)	Specifies what level of information, if any, is logged. Possible values are: off information warning error verbose **This setting is included in the default MSI file.** The default value is information.
Authentication & Privilege	GPDeployment	Specifies whether the computer is joined to the zone where the computer was pre-created. This setting is used only during installation and does not have a corresponding registry key. Possible values are: 0 - The computer is not joined to the zone. 1 - The computer is joined to the zone. **This setting is included in the default transform (MST) file.** To use it, you must execute the MST file when you execute the default MSI file. The default value is 1, meaning that the pre-created computer is joined to the zone.

Editing the Default Transform (MST) File

The default transform file, Group Policy Deployment.mst, enables you to specify registry key settings that are different from the default settings that are defined in the MSI file. You can use the Group Policy Deployment.mst file to customize a silent installation for a specific environment.

If you want to customize the agent settings for your environment, you should edit the Group Policy Deployment.mst file before executing the command to perform a silent installation. If you want to use the default settings specified in the MSI file, you can skip this section and go directly to Installing silently from the command line.

You must use the Orca MSI editor to edit the Group Policy Deployment.mst file. Orca is one of the tools available in the Windows SDK. If you do not have the Windows SDK or Orca installed on your computer, you can download and install it from this location: http://msdn.microsoft.com/en-us/library/aa370557(v=vs.85).aspx.

To edit the default MST file:

1. In the Agent folder in the Centrify download location, create a backup copy of the default Group Policy Deployment.mst file.

2. Open a Command Prompt window and execute the following command to launch Orca:

   Orca.exe

3. In Orca, select **File > Open** and open the Centrify Agent for Windows64.msi file located in the Agent folder in the Centrify download location.
4. In Orca, select **Transform > Apply Transform**.
5. In Orca, navigate to the Agent folder in the Centrify download location and open Group Policy Deployment.mst. The file is now in transform edit mode, and you can modify data rows in it.
6. In the Orca left pane, select the Property table. Notice that a green bar displays to the left of "Property" in the left pane. This indicates that the Property table will be modified by the MST file. The right pane displays the properties that configure registry keys when you execute the command to install the agent using the MSI file. Notice that the last property in the table, GPDeployment, is highlighted in a green box. This indicates that the GPDeployment property will be added to the MSI file by the MST file.
7. In the right pane, edit or add properties as necessary to configure registry keys for your environment.

REG_MAX_FORMAT	Sets the MaxFormat registry key to specify the color depth of sessions recorded by the agent. The color depth affects the resolution of the activity recorded and the size of the records stored in the audit store database when you have video capture auditing enabled. In the MSI file Property table, you can set the color depth to one of the following values: #0 to use the native color depth on an audited computer. #1 for a low resolution with an 8-bit color depth. #2 for medium resolution with a 16-bit color depth. #4 for highest resolution with a 32-bit color. The default value is #1. To edit this property, double-click the Value column and type a new value.
REG_DISK_CHECK_THRESHOLD	Sets the DiskCheckThreshold registry key to specify the minimum amount of disk space that must be available on the disk volume that contains the offline data storage file. In the MSI file Property table, the default value is #10, meaning that at least 10% of the disk space on the volume that contains the offline data storage file must be available. You can change the percentage required to be available. To edit this property, double-click the Value column and type a new value.
REG_SPOOL_DIR	Sets the SpoolDir registry key to specify the offline data storage location. The folder location you specify will be where the agent saves data when it cannot connect to a collector. To add a this property to the transform file, right-click anywhere in the property table, then select **Add Row**.
REG_INSTALLATION_ID	Sets the InstallationId registry key to specify the unique global identifier (GUID) associated with the installation service connection point. This property is not required if you are using the Installation group policy to identify the audit installation to use. If you are not using group policy to identify the audit installation, you can add a this property to the transform file. Right-click anywhere in the property table, then select **Add Row** to add the property and value to the file.
REG_LOG_LEVEL_DA	Sets the LogLevel registry key to specifies what level of information, if any, is logged. Possible values are: off information warning error verbose The default value is information. To edit this property, double-click the Value column and type a new value.

8. After you have made the necessary modifications, select **Transform > Generate Transform** to save your modifications to the default MST file. Be sure to save the MST file in the same folder as the MSI file. If the MST and MSI files are in different folders, the MST file will not execute when you execute the MSI file.

The MST file is now ready to be used as described in Installing silently from the command line.

Installing Silently from the Command Line

If you want to perform a "silent" or unattended installation of the Centrify Agent for Windows, you can do so by specifying the appropriate command line options and Microsoft Windows Installer (MSI) file to deploy.

Before running the installation command, you should verify the computers where you plan to install meet the prerequisites described in Verify prerequisites. If the prerequisites are not met, the silent installation will fail. You should have also completed the following tasks:

- Installed and configured the SQL Server management database and the SQL Server audit store database.
- Installed and configured one or more collectors.
- Configured and applied the Centrify DirectAudit Settings group policy that specifies the installation name.

You can use similar steps to install the Centrify Common Component using the Centrify Common Component64.msi file before you install the agent. If you install the common component first, information about the agent installation is recorded in a log file for troubleshooting purposes. However, you are not required to install the common component separately from the agent.

To install the Centrify Agent for Windows silently:

1. Open a Command Prompt window or prepare a software distribution package for deployment on remote computers.
2. Run the installer for the Centrify Agent for Windows package for a 64-bit architecture with the appropriate command line options. For example, to install the Centrify Common Component on a computer with 64-bit architecture, run the following command: msiexec /i "Centrify Common Component64.msi" /qn If you want to enable both auditing and access control features on a computer with a 64bit operating system and use the values defined in the Group Policy Deployment.mst file, you would run the following command: msiexec /i "Centrify Agent for Windows64.msi" /qn TRANSFORMS="Group Policy Deployment.mst"

Installing from a Central Location by Using Group Policy

You can use a Group Policy Object (GPO) to automate the deployment of Centrify Agent for Windows. Because automated installation fails if all the prerequisites are not met, be sure that all the computers on which you intend to install meet the requirements described in Verify prerequisites.

You can use similar steps to install the Centrify Common Component using the Centrify Common Component64.msi file before you install the agent. If you install the common component first, information about the agent installation is recorded in a log file for troubleshooting purposes. However, you are not required to install the common component separately from the agent.

In most cases, you can use the default agent settings defined in the Group Policy Deployment.mst transform file. If you want to modify the default settings prior to installation, see the instructions in Installing silently by using the Microsoft Windows Installer.

To create a Group Policy Object for the deployment of Centrify Agents for Windows:

1. Copy the Centrify Agent for Windows64.msi and Group Policy Deployment.mst files to a shared folder on the domain controller or a location accessible from the domain controller. When you select a folder for the files, right-click and select **Share with > Specific people** to verify that the folder is shared with Everyone or with appropriate users and groups.
2. On the domain controller, click **Start > Administrative Tools > Group Policy Management**.
3. Select the domain or organizational unit that has the Windows computers where you want to deploy the Centrify Agent, right-click, then select **Create a GPO in this domain, and Link it here**. For example, you might have an organizational unit specifically for Centrify-managed Windows computers. You can create a group policy object and link it to that specific organizational unit.
4. Type a name for the new Group Policy Object, for example, Centrify Agent Deployment, and click **OK**.
5. Right-click the new Group Policy Object and click **Edit**.
6. Expand **Computer Configuration > Policies > Software Settings**.
7. Select **Software installation**, right-click, and select **New > Package**.
8. Navigate to the folder you selected in Step 1, select the Centrify Agent for Windows64.msi file, and click **Open**.
9. Select **Advanced** and click **OK**.
10. Click the **Modifications** tab and click **Add**.
11. Select the Group Policy Deployment.mst file, click **Open**, and click **OK**.
12. Close the Group Policy Management Editor, right-click the Centrify Agent Deployment group policy object, and verify that **Link Enabled** is selected.

By default, when computers in the selected domain or organizational unit receive the next group policy update or are restarted, the agent will be deployed and the computer will be automatically rebooted to complete the deployment of the agent.

If you want to test deployment or deploy immediately, you can open a Command Prompt window to log on to a Windows client as a domain administrator and

force group policies to be updated immediately by running the following command:

gpupdate /force

After installation, all of the registry settings that were specified in the MSI and MST files are configured. If you need to change any of the default agent settings, open the DirectAudit Agent Control Panel or the Registry Editor.

For more information about how to configure and use Group Policy Objects, see the documentation on the Microsoft Windows website.

Enabling or Disabling Auditing on Windows Computers

You enable or disabling auditing for a Windows computer by adding or removing the audit and monitoring service from the agent configuration.

To enable auditing on a Windows computer:

- Use the Agent Configuration wizard to configure the Centrify Agent for Windows to connect to the Centrify Audit & Monitoring Service.

  The agent configuration wizard runs automatically after you've installed the Centrify Agent for Windows.

To disable auditing on a Windows computer:

- In the Centrify Agent Configuration window, select Centrify Audit & Monitoring Service and click **Remove**.

To enable or disable video capture editing for an entire installation, see Enabling or disabling video capture auditing . To enable or disable auditing on a per-user basis you can use a group policy and the audited user list and non-audited user lists. For details, see the Group Policy Guide.

Enabling or Disabling Auditing on Linux and UNIX Computers

After you install the agent, you can enable auditing with the dacontrol command. The dacontrol command links all shells to the cdash shell wrapper by way of NSS. When a user opens a terminal, cdash is automatically loaded instead of the user's shell, then cdash loads the appropriate shell for the user and begins auditing the session.

You can also choose to enable video capture editing for an installation but disable it for specific computers. You disable or enable video capture auditing for a specific computer or set of computers by using group policy settings or by modifying the agent.video.capture setting. For details, see the *Group Policy Guide* or the *Configuration and Tuning Reference Guide*.

Shell or Terminal Window Auditing

To enable auditing on a Linux or UNIX computer:

1. Log on as a user with root privileges.
2. Run dacontrol with the -e option:

   dacontrol -e
3. Run dacontrol again to verify that auditing has been enabled or run dainfo.

   For example, the output of the dacontrol command shows something like this:

   dacontrol --query

   This machine has been configured through group policy to use installation 'DefaultInstallation'

   DirectAudit NSS module: Active

   DirectAudit is not configured to audit individual commands.

   When you enable auditing, the NSS module shows as active. You can also see if auditing is enabled or not for a system in the Audit Manager console.

After you enable auditing on a Linux or UNIX computer, you can control whether the auditing of shell activity applies for all users or for selected users by using role assignments. If auditing is enabled and the agent is not running, users with an active role assignment that requires logging are not allowed to log in.

For more information about configuring and assigning roles, see the *Administrator's Guide for Linux and UNIX*.

To disable auditing on a Linux or UNIX computer:

1. Log on as a user with root privileges.
2. Run dacontrol with the -d option or the --disable option:

   dacontrol -d

   dacontrol --disable
3. Run dacontrol again to verify that auditing has been disabled or run dainfo.

   For example:

   dacontrol --query

   This machine has been configured through group policy to use installation 'DefaultInstallation'

   DirectAudit NSS module: Inactive

   DirectAudit is not configured to audit individual commands

   When you disable auditing, the NSS module shows as inactive. You can also see if auditing is enabled or not for a system in the Audit Manager console.

Linux Desktop Auditing

In addition to shell auditing, for some Linux systems you can also enable desktop auditing. When desktop auditing is enabled, the user's entire screen is continuously monitored to record all graphical interactions. More specifically, desktop auditing captures the following:

- The application name and window title when the user switches the focus to that application. For example, if a user opens a web browser or a terminal window.
- Changes to the application window title that currently has focus. For example, if a user opens a web browser and goes to a new web page, desktop auditing records the title of a web page.

The supported platforms for Linux desktop auditing are as follows:

- RHEL 6, 7, and 8 with GNOME v3
- CentOS 6, 7, and 8 with GNOME v3

Linux sessions must be running X as the primary display manager (not Wayland).

Linux desktop auditing requires shell session auditing.

To enable desktop auditing on a Linux computer:

1. Log on as a user with root privileges.
2. Run dacontrol with the -x option or the --desktop-audit option:

   dacontrol -x

   dacontrol --desktop-audit

   To enable both shell and desktop auditing at the same time, use both the -e and -x options:

   dacontrol -e -x
3. Run dainfo to verify that desktop auditing has been enabled.

   For example, the relevant information from the dainfo command looks like this:

   Pinging adclient: adclient is available
   Daemon status: Online
   Current installation: 'DirectAudit' (configured locally)
   Current collector: test.acme.com:5063:HOST/test.acme.com@acme.com
   DirectAudit NSS module: Active
   ... DirectAudit desktop auditing: Enabled
   User (root) audited status: Yes

   When you enable auditing, the desktop auditing module shows as Enabled. You can also see if auditing is enabled or not for a system in the Audit Manager console.

To disable desktop auditing on a Linux computer:

1. Log on as a user with root privileges.
2. Run dacontrol with the -z option or the --no-desktop-audit option:

   dacontrol -z

   dacontrol --no-desktop-audit
3. Run dainfo to verify that desktop auditing has been disabled.

   For example, the relevant information from the dainfo command looks like this:

   Pinging adclient: adclient is available
   Daemon status: Online

   Current installation: 'DirectAudit' (configured locally)
   Current collector: test.acme.com:5063:HOST/test.acme.com@acme.com

DirectAudit NSS module: Inactive
... DirectAudit desktop auditing: Disabled
User (root) audited status: No

When you disable auditing, the desktop auditing module shows as Disabled. You can also see if auditing is enabled or not for a system in the Audit Manager console.

Installing an Centrify Agent for Unix/Linux

You can install the auditing services for Linux or UNIX computers interactively the agent installation script, install.sh. If you want to run the installation script silently or use a native package manager to install UNIX agents, see Installing the UNIX agent on remote computers.

The steps in this section describe how to install interactively using the install.sh script, which automatically installs platform specific software packages and invokes the proper installation mechanism and options for a computer's operating system.

To install the agent using the installation script:

1. Log on as a user with root privileges.
2. Mount the cdrom device using the appropriate command for the local computer's operating environment, if necessary.

   Note:If you are not using the CD, verify the location and go on to the next step.
3. Change to the appropriate directory.

   For example, to install on an AIX computer from the Centrify CD or ISO file, change to the UNIX directory:

   cd Agent_Unix22
4. Run the installer and respond to its questions:

   ./install.sh

   If there is an installation with the name DefaultInstallation, the UNIX agent uses it by default. If you are using an installation with a name other than DefaultInstallation, you must identify the installation by using dacontrol or group policy after installing the agent. For more information, see Checking the status of the UNIX agent.
5. After installing the package, use dainfo to verify that the agent is installed and running. You should see output similar to the following that indicates the agent is Online:

   Pinging adclient: adclient is available

   Daemon status: Online

   Current installation: 'PistolasSF' (configured locally)

   Current collector: DC2008r2-LG.pistolas.org:5063:HOST/dc2008r2-lg@PISTOLAS.ORG ...

   If the output of dainfo indicates that the agent is Offline or that auditing is not enabled, verify your network connections and try restarting the auditing service or run the command to enable auditing manually as described in Enabling or disabling auditing on Linux and UNIX computers.

You must adjust the disk space requirements higher if you allocate a large amount of offline storage to use when none of the collectors servicing the audit store can be reached. This and other parameters are in a text file named centrifyda.conf in /etc/centrifyda on each audited computer that has the UNIX agent installed. For more information about setting configuration parameters, see Configuring the UNIX agent off-line database. For information about all of the configuration parameters available to customize auditing, see the *Configuration and Tuning Reference Guide*.

Enabling or Disabling Video Capture Auditing

In most cases, you decide whether to enable video capture auditing when you create a new installation. You can, however, choose to enable or disable video capture auditing for an installation at any time. For example, you might enable full video capture auditing of user activity during your initial deployment and later find that you are capturing user activity that is of no interest or requires too much database management to store. Conversely, you might initially decide not to enable video capture and later discover that you want to record complete information about user activity when users run privileged commands or open certain applications.

You can also choose to enable video capture editing for an installation but disable it for specific computers. You disable or enable video capture auditing for a specific computer or set of computers by using group policy settings or by modifying the agent.video.capture setting. For details, see the *Group Policy Guide* or the *Configuration and Tuning Reference Guide*.

For information about enabling or disabling auditing on Windows or Linux/UNIX computers, see Enabling or disabling auditing on Windows computers and Enabling or disabling auditing on Linux and UNIX computers

To enable or disable video capture auditing for an installation:

1. In the Audit Manager console, right-click the installation name, then select **Properties**.
2. Click the **Audit Options** tab.
3. Select **Enable video capture auditing of user activity** if it is not selected to start capturing a visual record of all user activity when users perform tasks using a role that is configured to be audited.

   Deselect this option to stop all video capture auditing. If you disable video capture auditing, you will not be able to replay session activity.
4. Click **OK** or **Apply**.

Installing Additional Audit Manager or Audit Analyzer Consoles

If you need to make Audit Manager or Audit Analyzer consoles available to other users, you can install additional consoles on other computers. For example, install Audit Analyzer on computers used by auditors in your organization.

Delinea

Checklist for Auditing Systems Outside of Active Directory

Here is the overall process for auditing a computer that isn't joined to Active Directory, including links to documented procedures.

	Create the audit installation	
1	For the audit store that includes the collector that you will enroll to the Privileged Access Service, edit the audit store scope so that it includes the following: The site or subnet that the collector is in The IP address or subnet of the system to be audited (the one that isn't in Active Directory)	Creating a new installation
	Add the audit installation to the Admin Portal and enable auditing	
2	Install a connector on a Windows computer in the Active Directory domain Note: For now, do not install a connector on the same computer as a collector.	"How to install a connector" in the Privileged Access Service help (docs.centrify.com)
3	In the Admin Portal, enable auditing for the audit installation.	"Enabling auditing for remote sessions" in the Privileged Access Service help (docs.centrify.com)
4	Verify the connector status in the Admin Portal. **Note**: If your deployment is across multiple Active Directory forests or you have multiple DirectAudit installations, your deployment will include multiple cloud connectors. In this kind of deployment, you should configure each non-Active Directory system to use only the cloud connectors that are in the same Active Directory forest as the desired DirectAudit installation. You can configure which connectors should be used in the system's Connector settings in the Admin Portal. For details, see the "Selecting the connectors to use" topic in the Privileged Access Service help (docs.centrify.com).	"Reference content - Connector configuration program" in the Privileged Access Service help (docs.centrify.com)
	Configure the collector	
5	On the computer where the collector is or will be, install the Centrify Client and enroll the computer in the Privileged Access Service. The collector needs to be joined to Active Directory and enrolled in the Privileged Access Service.	"Installing and using the Centrify Client for Windows" in the Privileged Access Service help (docs.centrify.com)
6	Install a new collector or reconfigure an existing collector so that the collector receives audit data according to the cloud settings.	Configure the audit collector service
	Configure the computer to be audited	
7	In the Admin Portal, download the Centrify Client installers and get an enrollment code	"Installing and using the Centrify Client for Windows" in the Privileged Access Service help (docs.centrify.com) "Enrolling and managing computers using Centrify Client for Linux" in the Privileged Access Service help (docs.centrify.com) "Enrolling a computer" in the Privileged Access Service help (docs.centrify.com)
8	In the Admin Portal, make sure that the user account you'll use to run the installer has the permissions to enroll the system.	"Admin Portal administrative rights" in the Privileged Access Service help (docs.centrify.com)
		On the computer to be audited, make sure that its

9	On the computer to be audited, make sure that its DNS setting are set so that it can contact and be contacted by the collector computer.	DNS settings are set so that it can contact the collector computer by its fully qualified domain name (FQDN).
10	Install the client and enroll the computer in the Privileged Access Service.	"Installing and using the Centrify Client for Windows" in the Privileged Access Service help (docs.centrify.com) "Enrolling and managing computers using Centrify Client for Linux" in the Privileged Access Service help (docs.centrify.com)
11	In the Admin Portal, verify the enrollment.	In the Admin Portal, go to Resources > Systems to verify the enrollment status.
12	Install the audit client package(s): Windows: Install the Windows audit package. Linux: First install the OpenSSL package, and then install the Linux audit package..	"Downloading the audit packages for the Centrify Clients" in the Privileged Access Service help (docs.centrify.com)
13	In Audit Manager, verify that the computer is being audited.	Managing audited computers and agents

Delinea

Auditing Systems That are Inside a DMZ

If you have Windows or UNIX/Linux systems that are deployed inside of a networking DMZ, you can audit those systems without having to set up a separate audit installation.

Organizations often use a DMZ to host a group of systems in a section of the corporate network in between the intranet and the public internet access. Firewall settings define the perimeter of the DMZ; the firewall helps limit access to internal networks from the outside network.

In order to audit systems inside of a DMZ, the following must be true for your deployment:

- All the Windows or UNIX/Linux systems in the DMZ are joined to the DMZ domain (for example, acme.dmz).
- You've already set up the audit installation in your main domain for your organization (for example, acme.corp) and you're auditing systems in that domain.
- The SQL Server that hosts the audit databases is also joined to the main domain.
- There's either no Active Directory trust between the main and DMZ domains or there's a one-way trust where the DMZ domain trusts the main domain (for example, acme.dmz trusts acme.corp).
- All the audit administrator and auditor accounts belong to the main domain.

Before you go to set up auditing on DMZ systems, be sure to do the following:

- Deploy at least one audit collector on a system that's joined to the DMZ domain. This is because an audited system can only look for audit collectors in its own forest.
- Configure the SQL Server to use mixed-mode authentication. This is because a collector in a DMZ cannot authenticate with SQL Server in the main domain using Windows authentication.
- Configure the necessary firewall exceptions for the SQL Server deployed in the main domain so that the audit collector in the DMZ can connect to the SQL Server. This includes the firewall exceptions for the SQL Server listener port as well as other ports, such as UDP 1434 (which is used by the SQL browser service).

To audit systems in a DMZ:

1. Prepare the audit store:

    1. Set up an audit store that contains the audited data for the systems in the DMZ. You can either create a new audit store or modify an existing one so that the audit store scope includes the sites or subnets of the systems in the DMZ.
    2. Add a new audit store database to the DMZ audit store and mark the database as active.

    For more information, see Creating the first audit store.

2. Prepare the SQL authentication account:

    1. In Audit Manager, right-click the audit store database that you just created and select **Properties**.
    2. In the Advanced tab, under the Allowed incoming collectors, click **Add**.
    3. For the authentication, select **SQL Server authentication**. Select an existing account or click the list to create a new SQL Login account.

    This SQL Login account is what the collectors in the DMZ domain will use to authentication with the SQL Server in the main domain. As a best practice, it's recommended to create a dedicated incoming collector account for all collectors in the DMZ.

3. Publish the audit installation information to the DMZ domain:

    - If there's a one-way trust between the DMZ and main domains:
        1. In Audit Manager, right click the installation name and click **Properties**.
        2. In the Publication tab, click **Add**.
        3. Select an OU or container in the DMZ domain to which you'll publish the audit installation information. Click **OK** to continue, and click **OK** again to close the dialog box and publish the audit installation information to the DMZ.

    For more information, see Publishing installation information.

    - If there's no trust between the DMZ and main domains:
        1. In Audit Manager, right-click the installation name and click **Properties**.
        2. In the Publication tab, click **Export** to export the audit installation information to an LDIF file.
        3. Provide the LDIF file to the Active Directory administrator of the DMZ domain and request that they manually import the file into an OU or container in the DMZ domain. They can import the LDIF file using the LDIFDE.exe utility.

For more information, see Exporting installation information.

4. Install a collector on at least one Windows system in the DMZ:

    1. Run the Collector Configuration wizard, and select the audit installation and specify the port number.
    2. In the Authentication type screen, select **SQL Server authentication** and enter the credentials for the SQL authentication account that you created earlier.
    3. Click **Test Connection** to ensure that the credentials work and the SQL Server is reachable.
    4. Finish the rest of the wizard. If there are any warnings when validating the permissions, you can safely ignore them.

    If you login to the collector computer as a user from your DMZ domain, that user will most certainly not have the permissions to connect to the audit installation and, as a result, the Collector Configuration wizard (which runs in context of the logged-in user) may fail to validate certain permissions and show warning messages instead.

    For details about configuring a collector, see Installing the audit collectors.

5. Install and configure the agent on the systems in the DMZ.

    For details, see Installing the Centrify Agent for Windows.

Managing an Installation

This section describes how to secure and manage an audit installation after the initial deployment of Centrify software. It includes tasks that are done by users assigned the Master Auditor role for an installation and users who are Microsoft SQL Server database administrators.

Topics available:

Securing an Installation

For production deployments, you can take the following steps to secure the installation:

- Use the Installation group policy to specify which installation agents and collectors are part of. By enabling the Installation group policy you can prevent local administrators from configuring a computer to be part of an unauthorized installation.
- Configure a trusted group of collectors to prevent a hacker from creating a rogue collector to collect data from agents.
- Configure a trusted group of agents to prevent a hacker from performing a Denial of Service attack on the collector and database by flooding a collector with bogus audit data.
- Encrypt all data sent from the collector to the database.

Before you can follow these steps to secure an installation, you must have access to an Active Directory user account with permission to create Active Directory security groups, enable group policies, and edit Group Policy Objects.

To secure an installation using Windows group policy:

1. Open the Group Policy Management console.
2. Expand the forest and domains to select the Default Domain Policy object.
3. Right-click, then click **Edit** to open Group Policy Management Editor.
4. Expand **Computer Configuration > Policies > Centrify Audit Settings**, then select **Common Settings**.
5. Double-click the **Installation** policy in the right pane.
6. On the Policy tab, select **Enabled**.
7. Click **Browse** to select the installation you want to secure, then click **OK**.
8. Click **OK** to close the Installation properties.

Securing an Audit Store with Trusted Collectors and Agents

By default, audit stores are configured to trust all audited computers and collectors in the installation. Trusting all computers by default makes it easier to deploy and test auditing in an evaluation or demonstration environment. For a production environment, however, you should secure the audit store by explicitly defining the computers the audit store can trust.

You can define two lists of trusted computers:

- Audited computers that can be trusted.
- Collector computers that can be trusted.

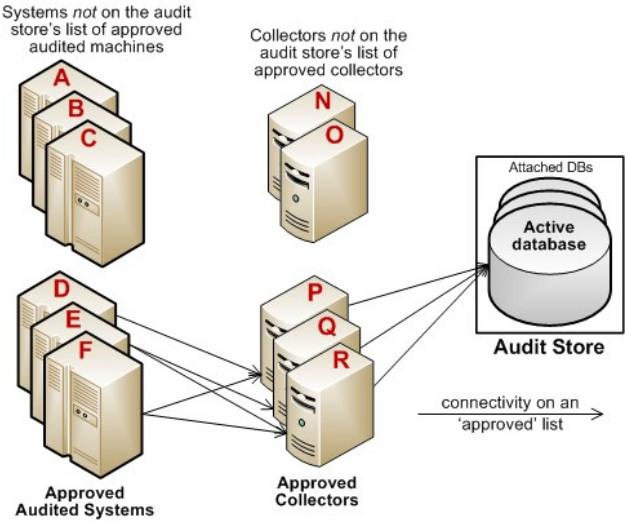
To secure an audit store:

1. Open the Audit Manager console.
2. Expand the installation and Audit Stores nodes.
3. Select the audit store you want to secure, right-click, then select **Properties**.
4. Click the **Advanced** tab.
5. Select **Define trusted Collector list**, then click **Add**.
6. Select a domain, click **OK**, then search for and select the collectors to trust and click **OK** to add the selected computers to the list.

   Only the collectors you add to the trusted list are allowed to connect to the audit store database. All other collectors are considered untrusted and cannot write to the audit store database.
7. Select **Define trusted Audited System list**, then click **Add**.
8. Select a domain, click **OK**, then search for and select the audited computers to trust and click **OK** to add the selected computers to the list.

   Only the audited computers you add to the trusted list are allowed to connect to the trusted collectors. All other computers are considered untrusted and cannot send audit data to trusted collectors.
9. Click **OK** to close the audit store properties dialog box.

The following example illustrates the configuration of trusted collectors and trusted audited computers.



In this example, the audit store trusts the computers represented by P, Q, and R.Those are the only computers that have been identified as trusted collectors in the audit store Properties. list. The audit store has been configured to trust the audited computers represented by D, E, and F. As a result of this configuration:

- Audited computers D, E, and F only send audit data to the trusted collectors P, Q, and R.
- Trusted collectors P, Q, and R only accept audit data from the trusted audited computers D, E, and F.
- The audit store database only accepts data for its trusted collectors P, Q, and R, and therefore only stores audit data that originated on the trusted audited computers D, E, and F.

Disabling a Trusted List

After you have added trusted collectors and audited computers to these lists, you can disable either one or both lists at any time to remove the security restrictions. For example, if you decide to allow auditing data from all audited computers, you can open the audit store properties, click the Advanced tab, and deselect the **Define trusted Audited System list** option. You don't have to remove any computers from the list. The audit store continues to only accept data from trusted collectors.

Using Security Groups to Define Trusted Computers

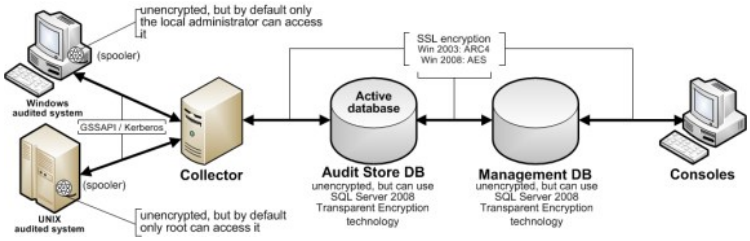
You can use Active Directory security groups to manage trusted computer accounts. For example, if you create a group for trusted audited computers and a group for trusted collectors, you can use those groups to define the list of trusted collectors and audited computers for the audit store. Any time you add a new computer to one of those groups, thereafter, it is automatically trusted, without requiring any update to the audit store properties.

Securing Network Traffic with Encryption

The last step in securing an installation is to secure the data collected and stored through encryption. The following summarizes how data is secured as it moves from component to component:

- Between an audited computer and the spooler that stores the data locally when no collectors are available, audit data is not encrypted. Only the root user or local Administrator account can access the data by default.
- Between the audited computer's data collection service (dad on UNIX or wdad on Windows) and the collector, data is secured using Generic Security Services Application Program Interface (GSSAPI) with Kerberos encryption.
- Between the collector and the audit store database, data can be secured using Secure Socket Layer (SSL) connections and ARC4 or AES encryption if the database is configured to use SSL connections.
- Between the audit store and management databases, data can be secured using Secure Socket Layer (SSL) connections and ARC4 or AES encryption if the database is configured to use SSL connections.
- Between the management database and the Audit Manager console, data can be secured using Secure Socket Layer (SSL) connections and ARC4 or AES encryption if the database is configured to use SSL connections.

The following illustration summarizes the flow of data and how network traffic is secured from one component to the next.



Enabling Secure Socket Layer (SSL) communication

Although the database connections can be secured using SSL, you must configure SSL support for Microsoft SQL Server as part of SQL Server administration. You must also have valid certificates installed on clients and the database server. If you are not the database administrator, you should contact the database administrator to determine whether encryption has been enabled and appropriate certificates have been installed. For more information about enabling SSL encryption for SQL Server and installing the required certificates, see the following Microsoft support article:

https://support.microsoft.com/en-us/help/316898/how-to-enable-ssl-encryption-for-an-instance-of-sql-server-by-using-mi

Enabling Encryption for Microsoft SQL Server Express

If you use Microsoft SQL Server Express, encryption is turned off by default. To secure the data transferred to the database server, you should turn encryption on.

To enable encryption for each audit store and management database:

1. Log on to the computer hosting an audit store or management database with an account that has database administrator authority.
2. Open **SQL Server Configuration Manager**.
3. Select the SQL Server Network Configuration node, right-click **Protocols for DBINSTANCE**, then select **Properties**.
4. On the **Flags** tab, select **Yes** for the **Force Encryption** option, then click **OK** to save the setting.

Using a Service Account for Microsoft SQL Server

When you install Microsoft SQL Server, you specify whether to use Windows authentication or a mix of Windows and SQL Server authentication. You also specify the accounts that the database services should use. By default, system accounts are used. If SQL Server uses a domain user account instead of a system account, you should ensure that the account has permission to update the SQL Server computer object in Active Directory. If the account has permission to update the computer where SQL Server is running, SQL Server can publish its service principal name (SPN) automatically. Getting the correct service principal name is important because Windows authentication relies on the SPN to find services and audit and monitoring service uses Windows authentication for console-to-audit management database connections. If the SPN is not found, the connection between the console and audit management database fails.

The audit management database-to-audit store connection and the collector-to-audit store connection can use either Windows authentication or SQL Server authentication. If SQL Server authentication is used, it does not matter whether the SQL Server instance uses a system account or a service account.

If you have configured SQL Server to use Windows authentication only, be sure that the Windows account is allowed to connect to the audit management database and to the audit store database.

If the domain user account running SQL Server services does not have permission to update the computer object, see the following Microsoft knowledge base article for information about how to manually register the SPN for SQL Server:

https://support.microsoft.com/en-us/help/909801/how-to-make-sure-that-you-are-using-kerberos-authentication-when-you-c

Configuring Selective Auditing

By default, the agent captures activity for all users on audited computers, but you can limit auditing to specified users. If you are using authentication and privilege elevation, you can control auditing by configuring role definitions with different audit requirements then assigning those role definitions to different sets of Active Directory users.

If you are using the Centrify Audit & Monitoring Service without access management:

- You can use group policies to specify which Windows users to audit and which Windows users should not be audited.

  For information about configuring group policies to customize auditing, see the *Group Policy Guide*.

- For UNIX users, you can use the dash.user.skiplist configuration parameter to specify the UNIX user accounts and Active Directory UNIX names that you don't want to audit.

  For more information about setting the dash.user.skiplist parameter, see the comments in the /etc/centrifyda/centrifyda.conf file. For information about all of the configuration parameters available to customize auditing, see the *Configuration and Tuning Reference Guide*.

Controlling Auditing by Using Group Policies

1. Open the Group Policy Management console.
2. Expand the forest and domains to select the Default Domain Policy object.
3. Right-click, then click **Edit** to open Group Policy Management Editor.
4. Expand **Computer Configuration > Policies > Centrify Audit Settings**, then select **Windows Agent Settings**.
5. Select the Audited user list policy and change the policy setting from Not Configured to **Enabled**, then click **Add** if you want to identify specific users to audit.

   When you enable this group policy, only the users you specify in the policy are audited. If this policy is not configured, all users are audited.
6. Select the Non-audited user list policy and change the policy setting from Not Configured to **Enabled**, then click **Add** if you want to identify specific users that should not be audited.

   When you enable this group policy, only the users you specify are not audited. If this policy is not configured, all users are audited. If you enable both the Audited user list and the Non-audited user list policies, the users you include in the Non-audited user list take precedence over the Audited user list.

The following table details the effect of choosing to enable the Audited user list policy, the Non-audited user list policy, or a combination of both policies.

Not configured	Not configured	No users are defined for either policy, so all users accessing audited computers are audited.
Not configured	Enabled	Only the users you specify in the Audited user list policy are audited. If you do not specify any users when you enable this policy, no users are audited.
Enabled	Not configured	Only the users you specify in the Non-audited user list are exempt from auditing. If you enable this policy but do not specify any users, no users are exempt from auditing. All users are audited.
Enabled	Enabled	If both policies are enabled, the non-audited user takes precedence over the audited list of users. If a user is specified in the audited list, that user is explicitly audited. If a user is specified in the non-audited list, that user is explicitly not audited. If the same user is specified in both lists or no users are specified for either policy, no users are audited because the non-audited user takes precedence.

Configuring agents to prefer collectors

If desired, you can specify that agents first use the collectors that are in the same site as the agent. You configure this option for each audit store.

For example, consider the following installation setup:

- One audit store
- Two sites (SantaClara and SanDiego)
- Two collectors in SantaClara, and two collectors in SanDiego

If you enable the option for the agents to prefer collectors in the same site as the agent, the agents in the SantaClara site use the collectors in that site, and the agents in the SanDiego site use the collectors there.

If for some reason all collectors in a site are down, the agents use collectors in another site or configured subnet.

Once an agent fails over and uses a collector in another site, the agent continues to use that collector until a rebinding occurs. You can do a rebinding with the dareload -b command. During the time that the agent is using a collector in another site, dadiag displays a warning message.

If your installation uses agents older than version 2017, those older agents ignore the collector preference setting.

Specify Agents Use Collectors in the Same Site

1. Open the Audit Manager console window.
2. Expand **Audit Stores**, and right-click the desired audit store and select **Properties**.
3. In the Audit Store Properties dialog box, click **Advanced**.
4. Select **Agents must prefer collectors in the same site as the agent**.

   By default, this option is not enabled.
5. Click **OK** to save the changes.

It may take several minutes for the changes to take effect, depending on Active Directory replication delays and policy sets.

Audit License Enforcement

Any time you open the Audit Manager console, Audit Analyzer console, or the session player, a background process determines the availability of audit licenses. Only the audited computers that are currently connected to a collector are included in the license count to determine license usage and compliance. Computers that have been previously audited and have data in the audit store database but are not currently connected to a collector and haven't been connected to a collector for over 45 days are not included in the license count.

As you increase the number of licenses in use, license enforcement is progressive. If the number of audited computers is less than 90% of the number of licenses you have purchased, there's no affect on any auditing features. If the number of audited computers is more than 90% of the licenses purchased, enforcement depends on the number of licenses in use:

- 90-100% of the licensing limit displays a warning message that you are close to over deployment, but you can continue to use all auditing features.
- 100-120% of the licensing limit displays a warning message that you must acknowledge by clicking **OK** when you open any console, after which you can resume using the console or session player.
- Over 120% of the licensing limit displays a warning message for 60 seconds when you open any console. If you see the 60 second warning message, use the License dialog box to add license keys to continue using auditing features.

You can contact Centrify to purchase additional licenses or remove some audited computers from the installation to bring the number of licenses used into compliance.

Agents and Licenses from Previous Versions

An installation can include agents and licenses from previous versions of Server Suite. For example, an installation might include a mix of UNIX and Windows agents from DirectAudit 2.x, or DirectAudit 3.x, or all new agents on the computers you want to audit.

Enabling Audit Notification on Windows

If you enable audit notification, users see a message informing them that their actions are being auditing when they log on. After you enable notification, the message is always displayed on audited computers if the session activity is being recorded.

> **Note**: This audit notification banner mentioned below displays on Windows systems. For notifying users on UNIX or Linux systems, the messages specified in the dash.prompt.message.file parameter apply. For details, see the *Configuration and Tuning Reference Guide*.

To enable audit notification for an installation:

1. In the Audit Manager console, right-click the installation name, then select **Properties**.
2. Click the **Notification** tab.
3. Select **Enable notification**.

   Deselect this option to turn off notification.
4. Click the browse button to locate and select a text file that contains the message you want to display.

   A notification message is required if you select the Enable notification option. The contents of the file you select are displayed below the file location. The maximum text file size is 30 KB.
5. Click the browse button to locate and select an image to appear as a banner across the top of the audit notification.

   Displaying a banner image is optional when you enable notification. The maximum image file size is 15 KB. For the best image display, use an image that is 468 pixels wide by 60 pixels high.

   > **Note**: Animated GIF files are not supported for use as audit notifications. If you do specify an animated GIF, the image displays as a static image.
6. Click **OK** or **Apply**.

   Users will see the notification message the next time they log in.
7. If you enable notification after you have deployed agents, update the local policy on the audited computers by running the following command:

   gpupdate /FORCE

Preventing Users from Reviewing or Deleting Sessions

By default, users can update the review status, add comments, and delete their own sessions if they have an audit role with the appropriate permissions. However, there are installationwide options to prevent any users from updating the review status or deleting their own sessions. These installation-wide options take precedence over audit role permissions for all users.

To prevent all users from updating the review status or deleting their own sessions in an installation:

1. In the Audit Manager console, right-click the installation name, then select **Properties**.
2. Click the **Audit Options** tab.
3. Select the appropriate settings for your installation.
    - Select **Do not allow any users to review their own sessions** if you to prevent all users from updating the review status or adding comments to their own sessions in Audit Analyzer.
    - Select **Do not allow any users to delete their own sessions** if you to prevent all users from deleting their own sessions in Audit Analyzer.
4. Click **OK** or **Apply**.

Adding an Installation

Although a single installation is the most common deployment scenario, you can configure multiple installations. For example, you can use separate installations to provide concurrent production and test-bed deployments or to support multiple administrative domains within your organization.

To create a new installation:

1. Open Audit Manager.
2. Select the root node, right-click, then select **New Installation**.
3. Follow the prompts displayed.

   The steps are the same as the first installation. For more information, see Creating a new installation.
4. Choose the appropriate installation for each collector using the Collector Configuration wizard.
5. Choose the appropriate installation for each agent using the Agent Configuration wizard.

Once you have multiple installations, you can choose which one each collector is part of using the Collector Configuration wizard. You can choose which installation each agent is part of using the Agent Configuration wizard. You can also configure collectors and agents using group policy.

> **Note**: Agents can communicate with a collector only if the agents and collector are in the same Active Directory forest.

Delegating Administrative Tasks for a New Installation

The account you use to create a new installation is the default administrator and Master Auditor with full control over the entire installation and the ability to delegate administration tasks to other Active Directory users or groups. You can grant permission to perform administrative tasks to other users by opening the Properties for each component, then clicking the Security tab.

Opening an Installation in a New Console

If you create multiple installations at the same site, you can select the installation name, right-click, then select **New Window From Here** to keep consoles for different installations separate from each other. Creating a new window for each installation can help you avoid performing operations on one installation that you intended to perform on another.

Closing an Installation

The Audit Manager console allows you to manage multiple installations. To remove the current installation from the console, but not physically remove the database or the information published to Active Directory, you can select the installation name, right-click, then select **Close**.

Publishing Installation Information

Centrify Audit & Monitoring Service publishes installation information to a service connection point (SCP) object in Active Directory so that audited computers and collectors can look up the information. If the published locations for multiple SCPs in the same installation are not the same, or if collectors cannot read from at least one of the published locations, the collectors are unable to determine which audit store is the best match for the sites and subnets, and so they do not attempt to connect to an audit store.

Permission to Publish to Active Directory

Only administrators who have been delegated permission to modify various attributes of the installation can publish those attributes to Active Directory.

At a minimum, you must have the Active Directory permission to Create serviceConnectionPoint objects on the container or organizational unit that you have identified for publishing installation information.

If you do *not* have Active Directory permission to modify the installation, the updates are kept in the audit management database, and a message is issued to notify you that the installation information could not be updated in Active Directory.

Synchronizing Installation Information

If you have an Active Directory account with permission to publish information about the installation, you can update the service connection point.

To publish the service connection point for an installation:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Publication** tab, then click **Synchronize** to publish the information.

   In a multi-forest or DMZ environment, this tab lists multiple Active Directory locations to which to publish.
4. Click **OK** to close the installation properties.

Exporting Installation Information

If you have an Active Directory account with permission to access installation information, you can export the service connection point that contains the installation information to a file in LDAP Data Interchange Format (LDIF). Exporting installation information can be useful if you want to add the domain for a perimeter network to an existing installation. After exporting installation information to a file, you can modify the file—for example, to use a different domain component—then import the modified file using the ldifde command.

To export and import installation information:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Publication** tab.
4. Select the Active Directory location, then click **Export**.
5. Select a file location and type a file name, then click **Save**.
6. Click **OK** to close the installation properties.
7. Use a text editor to modify the file, as needed.

   For example, you might use a different domain component—such as DC=dmz1,DC=ajax, DC=org in place of DC=internal, DC=ajax, DC=org—to differentiate between the perimeter and internal networks.
8. Import the modified file using a command similar to the following in a Command Prompt window:

   ldifde -i -f C:\Users\Administrator\Desktop\sample-dmz.ldif

Removing or Deleting an Installation

Before you can remove or delete an installation, you must do the following:

- Run the setup program to remove all agents and collectors and collector service connection points (SCPs).
- Detach and remove all audit store databases.
- Open the Installation Properties and click the **Publications** tab to make sure only one installation service connection point (SCP) is listed.

  Note: To remove service connection points on other sites, contact an administrator with publication permission on those sites.

To remove or delete an installation, select the installation in the Audit Manager console, right-click, then select **Remove** to open the Remove installation dialog box.

- Click **Remove** to remove the installation but *not* delete the management database from the SQL Server instance.
- Click **Delete** to remove the installation *and* delete the management database from the installation of SQL Server.

  Note: All the publications published to Active Directory are removed when you remove or delete an installation.

Managing Audit Store Databases

During the initial deployment, your installation only has one audit store database. As you begin collecting audit data, however, that database can quickly increase in size and degrade performance. Over time, an installation typically requires several Microsoft SQL Server databases to store the data being captured and historical records of session activity, login and role change events, and other information. As part of managing an installation, you must manage these databases to prevent overloading any one database and to avoid corrupting or losing data that you want to keep.

One of the biggest challenges in preparing and managing Microsoft SQL Server databases for storing audit data is that it is difficult to estimate the level of activity and how much data will need to be stored. There are several factors to consider that affect how you configure Microsoft SQL Server databases for auditing data, including the recovery method, memory allocation, and your backup and archiving policies.

The sections below provide guidelines for sizing and managing the Microsoft SQL Server databases you use for audit data. For more complete information about managing and configuring SQL Server, however, you should refer to your Microsoft SQL Server documentation.

Selecting a recovery model

Standard backup and restore procedures come in three recovery models:

- **Simple**—The Simple recovery model allows high-performance bulk copy operations, minimizes the disk space required, and requires the least administration. The Simple Recovery model does not provide transaction log backups, so you can only recover data to the point of the most recent full or differential backup. The default recovery model is Simple, but is not appropriate in cases where the loss of recent changes is not acceptable.
- **Full**—The Full recovery model has no work-loss exposure, limits log loss to changes since the most recent log backup, and provides recovery to an arbitrary time point. However, the Full recovery model uses much more disk space.
- **Bulk-logged**—The Bulk-logged recovery model provides higher performance and minimizes the log space used by disk-intensive operations, such as create index or bulk copy. With the Bulk-logged recovery model, you can only recover data to the point of the most recent full or differential backup. However, because most databases undergo periods of bulk loading or index creation, you can switch between Bulk-logged and Full recovery models to minimize the disk space used to log bulk operations.

When a database is created, it has the same recovery model as the **model** database. Although the Simple recovery model is the default, the Full and Bulk-Logged recovery models provide the greatest protection for data, and the Full recovery model provides the most flexibility for recovering databases to an earlier point in time. To change the recovery model for a database, use the ALTER DATABASE statement with a RECOVERY clause.

Regardless of the recovery model you choose, you should keep in mind that backup, restore, and archive operations involve heavy disk I/O activity. You should schedule these operations to take place in off-peak hours. If you use the Simple recovery model, you should set the backup schedule long enough to prevent backup operations from affecting production work, but short enough to prevent the loss of significant amounts of data.

Configuring the Maximum Memory for Audit Store Databases

Because Microsoft SQL Server uses physical memory to hold database information for fast query results, you should use a dedicated instance to store auditing data. Because SQL Server dynamically acquires memory whenever it needs it until it reaches the maximum server memory you have configured, you should set constraints on how much physical memory it should be allowed to consume.

The maximum server memory (max server memory) setting controls the maximum amount of physical memory that can be consumed by the Microsoft SQL Server buffer pool. The default value for this setting is such a high number that the default maximum server memory is virtually unlimited. Because of this default value, SQL Server will try to consume as much memory as possible to improve query performance by caching data in memory.

Processes that run outside SQL Server, such as operating system processes, thread stacks, socket connections and Common Language Runtime (CLR) stored procedures are not allowed to use the memory allocated to the Microsoft SQL Server buffer pool. Because those other processes can only use the remaining available memory, they might not have enough physical memory to perform their operations. In most casts, the lack of physical memory forces the operating system to read and write to disk frequently and reduces overall performance.

To prevent Microsoft SQL Server from consuming too much memory, you can use the following formula to determine the recommended maximum server memory:

- Reserve 4GB from the first 16GB of RAM and then 1GB from each additional 8GB of RAM for the operating system and other applications.
- Configure the remaining memory as the maximum server memory allocated for the Microsoft SQL Server buffer pool.

For example, if the computer hosting the Microsoft SQL Server instance has 32GB of total physical memory, you would reserve 4GB (from first 16 GB) + 1GB (from next 8 GB) + 1 GB (from next 8 GB) for the operating system, then set the Maximum server memory for Microsoft SQL server to 26GB (32GB – 4GB – 1GB – 1GB = 26).

For more information about how to configure Microsoft SQL Server maximum memory setting and other memory options, see the following Microsoft article:

http://msdn.microsoft.com/en-us/librms178067(v=sql.105).aspxary/ms178067(v=sql.105).aspx

You should configure the maximum memory allowed for the Microsoft SQL Server instances hosting audit store databases and the management database. However, this setting is especially important to configure on the Microsoft SQL Server instance hosting the active audit store database.

Using Transact-SQL to Configure Minimum and Maximum Memory

You can control the minimum and maximum memory that the SQL Server buffer manager uses by issuing Transact-SQL commands. For example:

```
sp_configure 'show advanced options', 1
reconfigure
go
sp_configure 'min server memory', 60
reconfigure
go
sp_configure 'max server memory', 100
reconfigure
go
```

For more information about configuring SQL Server and setting minimum and maximum server memory using T-SQL, see https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/server-memory-server-configuration-options?view=sql-server-2017

Estimating Database Requirements Based on the Data You Collect

To determine how auditing will affect database capacity, you should monitor a pilot deployment of 20 to 25 agents with representative activity to see how much data is produced daily. For example, some audited computers might have few interactive user sessions or only short periods of activity. Other audited computers might have many interactive user sessions or long sessions of activity on average.

During the pilot deployment, you want to the following information:

- How many interactive user sessions occur daily on each computer?
- How long do sessions last on average?
- What are the activities being captured, and what is the average size of each session being captured?
- How long do you need to store the captured data to balance performance and storage?
- What is the data retention period for audited data?

From the information you collect in the pilot deployment and the data retention policy for your organization, you can estimate the database size using the following guideline:

(number of agents) x (number of sessions per agent) x (average data size per session) x (retention days)

Results in the estimated size of the Microsoft SQL Server database for the number of days in the retention policy

For example, if an average session generated 100 KB in the database and the installation had 250 agents, 10 sessions per agent, and a six-month retention period (about 130 working days), the storage requirement for the audit store database would be 36.9 GB:

250 agents x 10 sessions/agent each day x 100 KB/session x 130 days = 32,500,000 KB

The following table shows examples of the data storage requirement in an installation with Windows agents, typical levels of activity with an average of one session per day on each audited computer, and the recovery mode set to Simple:

100	20 minutes	806 KB - low activity	79 MB	394 MB	10 GB
50	25 minutes	11.56 MB - high activity	578 MB	2.81 GB	73.36 GB
100	20 minutes	9.05 MB - high activity	905 MB	4.42 GB	115 GB

In this example, an installation with 100 Windows agents with low activity would require approximately 10 GB for the audit store database to keep audit data for 6 months. An increase in the number of interactive sessions, session length, or average session size would increase the database storage required.

If SQL Server requires more space to accommodate the new data, it expands the database file immediately, which can cause degraded performance. To reduce the effect of database expansion on performance, allocate sufficient space to support database growth. In addition, monitor database space and when space is low, schedule a database expand operation for an off-peak time.

Reducing Color Depth to Decrease Disk Usage

If you enable video capture auditing of user activity for an installation, the color depth setting affects the size of sessions stored in the audit store database. Depending on whether you want higher quality video playback or lower disk consumption, you can modify this setting. The growth rate is linear as you increase or decrease the color depth.

Based on a simulation of user activity, changing the color depth from 16-bit to 8-bit reduces disk space by 42%. Changing the color depth from 32-bit to 16-bit reduces disk space by 34 to 39%. If you can accept the lower quality video playback, changing the color depth from 32-bit to 8-bit reduces disk space by 62 to 65%.

Using SQL Server availability groups with multi-subnet failover for audit store databases

If you add an audit store database to a SQL Server availability group that has multiple subnet failover functionality, the SQL Server that hosts the management database must be SQL Server 2012 or above. This restriction applies only to availability groups that have multi-subnet failover configured.

For details about availability group multi-subnet failovers, see https://msdn.microsoft.com/en-us/library/hh213417.aspx#SupportAgMultiSubnetFailover.

Adding New Audit Store Databases to an Installation

When you first set up an installation, you also create the first audit store and audit store database. By default, that first database is the active database. As you begin collecting audit data, you might want to add databases to the audit store to support a rolling data retention policy and to prevent any one database from becoming a bottleneck and degrading performance.

Only one database can be the active database in an audit store at any given time. The computer hosting the active database should be optimized for read/write performance. As you add databases, you can change the older database from active to attached. Attached databases are only used for querying stored information and can use lower cost storage options.

> **Note**: A single instance of Microsoft SQL Server can host multiple databases. Those databases can support different versions of the agent.

Audit store databases have the following characteristics:

- A database can be active, attached, or detached.
- Only one database can be actively receiving audit data from collectors.
- A database cannot be detached while it is the active database.
- A database that was previously the active database cannot again be the active database.
- If a detached database contains parts of sessions presented to the Audit Analyzer, a warning is displayed when the auditor replays those sessions.

Rotating the Active Database

Database rotation is a management policy to help you control the size of the audit store database and the performance of database operations. There are several reasons to do database rotation:

- It is more difficult to manage one large database than multiple small databases.
- Performance is better with multiple small databases.
- Backing up, restoring, archiving, and deleting data all take significantly more time if you work with one large database.
- Database operations take very little time when you work with multiple small databases.

For audit and monitoring service, you can implement a database rotation policy by having the collector write data to a new database after a certain period of time. For example, the collector in site A writes data to the database siteA-2015-11 in November, then write data to database siteA-2015-12 in December and to the database siteA-2016-01 in January. By rotating from one active database to another, each database stays more compact and manageable.

Creating a New Database for Rotation

You can rotate from one active database to another at any time using the Audit Manager console.

To create a new database for rotation:

1. Open Audit Manager.
2. Expand the installation node, then expand Audit Stores and a specific audit store name.
3. Select **Databases**, right-click, then select **Add Audit Store Database** to create a new database.

   For details on setting up the database, see Creating the first audit store database.
4. Select the **Set as Active database** option so collectors start writing to the newly created database.

You can also use Centrify application programming interfaces (APIs) to write a script that automates the database rotation process. For API details and sample code, see the Centrify SDK documentation.

Database Archiving

To implement periodic archiving, add a new active database, leave one or more previous databases attached, and take the oldest database off-line for archiving.

Queries During Rotation and Archiving

If the database backup program supports online backups, the Audit Analyzer can still query the database while the backup is in progress. However, the backup program may block updates to the session review status. If the backup program does not support online backup, the database will be offline until the backup is complete.

Database Backups

You can back up a database whether it is attached to the audit store or detached from the audit store.

Reattaching a Restored Backup of a Database

If you need to query sessions from an older database that is offline and detached, you can restore the database from a backup and reattach it to your auditing installation. You might need to do this is your auditing installation is large, you do frequent database rotation, and you don't keep many databases attached and online.

Understand that after you restore a database from a backup, you need to fix a couple of settings in the database before you can reattach it to your auditing installation. During the backup operation, the database owner and trustworthy properties get set in such a way that prevents you from reattaching the database to your auditing installation unless you fix these properties.

To reattach a restored database to your auditing installation:

1. Run the following command to reset the database owner to [sa]:

   ALTER AUTHORIZATION ON DATABASE:: *<db_name>* TO [sa]
2. Run the following command to reset the trustworthy property:

   ALTER DATABASE *<db_name>* SET TRUSTWORTHY ON

   You can now reattach the database to your auditing installation.

Allowed Incoming Accounts

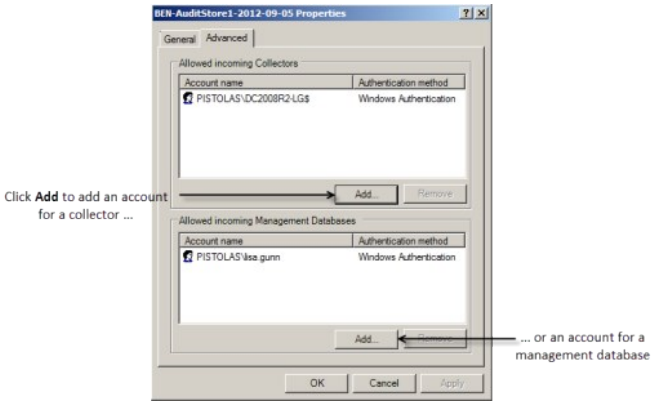
You can specify the accounts that are allowed to access the audit store database. By configuring these accounts, you can control which collector computers can connect to the audit store database and which management databases have access to the data stored in the audit store database.

Your account must have Manage SQL Login permission to configure the incoming accounts.

To configure allowed accounts:

1. Open Audit Manager.

2. Expand the installation node, then expand Audit Stores and select a specific audit store name.
3. Select a database under the audit store, right-click, then select **Properties**.
4. Click the **Advanced** tab.
5. Click **Add** to add a collector or management database account. For example:



6. Select an authentication type.
    - If you select Windows authentication, you can browse to select a computer, user, or group to add.
    - If you select SQL Server authentication, you can select an existing SQL Server login or create a new login.

Connections should use Windows authentication whenever possible. However, computers in an untrusted forest cannot connect to an audit management database using Windows authentication. To allow connections from an untrusted forest, add a SQL Server login account as the incoming account for the management database.

Detecting Data Tampering and Verifying Session Integrity

When you create your audit store database, you have the option to enable data integrity checking. Data integrity checking provides the ability to detect if auditing data has been tampered with.

For example, data integrity checking can detect if a user who has write privileges over the Audit Store database directly manipulates the audited session data by making a direct connection to the Microsoft SQL Server database.

Session data that is stored in audit store databases is typically accessible to database administrators and/or database owners in an unrestricted fashion. For these users with write privileges on the audit store database, it's fairly easy to tamper with data in such a way that it can help manipulate the outcome of an AQL query.

For example, someone could change the searchable tags in such a way so that the session is never returned by a query. Or, someone could remove suspicious activity from a recorded session, such as by changing the list of commands that are executed or changing the command output.

> **Note**: Data integrity checking cannot detect tampering if a database administrator deletes an entire session or database. Also, data integrity checking is not yet available for audit trail events.

Once you enable data integrity checking, you can do the following:

- Use the Audit Analyzer console or a PowerShell cmdlet to check the integrity of audited sessions.
- Determine if any data in the following tables has been modified and where it was modified:
    - Session
    - RawData
    - Command
    - SyscallCommand
    - SyscallFilemon
    - WashData
    - WashEvent
- Determine if any database rows belonging to an audited session were permanently deleted.

If you have not enabled an audit store database for data integrity checking and you try to check session integrity in Audit Analyzer, an error message appears.

Managing Audit Stores

An audit store is a collection of databases that contain audit data. All attached databases in the audit store are available to the audit management database. Typically each site has one audit store, but you can add audit stores as required for large or multi-site installations. For details, see Adding more audit stores to an installation.

Configuring Audit Store Scope

The scope of an audit store defines which audited computers send their audit data to the audit store, and which collectors are assigned to the audit store. The scope is a set of Active Directory sites and/or subnets. To configure the scope for an audit store, open its **Properties** page and select the **Scope** tab. To add a site, click **Add Site** and select the site from the list. To add a subnet, click **Add Subnet** and type a subnet address and mask.

Configuring Permissions for an Audit Store

To configure audit store security, open the audit store's **Properties** page and select the **Security** tab.

Only users with Change Permission permission on the audit store are allowed to modify the user rights on the Security tab of the audit store's Properties page.

The following table lists the rights that can be granted to active Directory users or groups, and the operations that the users granted such rights ("trustees") are allowed to perform.

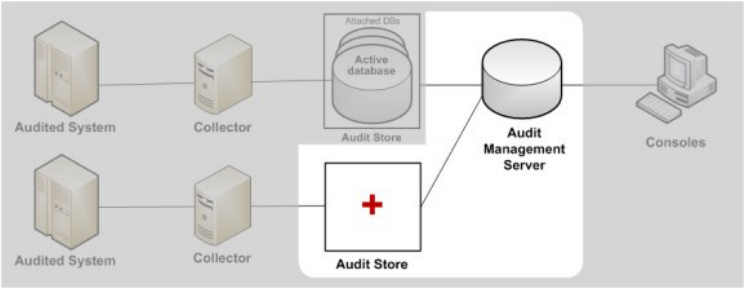
The audit store administrator by definition has all of these user rights (Full Control).

Full Control	All of the operations listed in the following rows of this table
Change Permissions	Modify permissions on this audit store
Modify Name	Modify display name for this audit store
Manage Scopes	Add a subnet or active Directory site Remove a subnet or active Directory site
Manage SQL Logins	Set the allowed incoming accounts for this audit store's databases Set the allowed incoming accounts for collectors
Manage collectors	Enable collector trusted group for this audit store Add collector to the trusted collector group in this audit store Remove collector from the trusted collector group in this audit store Remove disconnected collector record from this audit store
Manage Audited Systems	Enable audited computers trusted group for this audit store Add audited computer to the trusted audited computer group in this audit store Remove audited computer from the trusted audited computer list in this audit store Remove disconnected audited computer record from this audit store
Manage Databases	Add audit store database to this audit store Attach audit store database to this audit store Detach an audit store database from this audit store Change active database in this audit store Modify the display name of a version 2 audit store database
Manage Database Trace	Enable or disable database trace Export database trace

Adding More Audit Stores to an Installation

The audit store typically maps one-to-one with an Active Directory site. However, in some situations it is desirable to define the scope of an audit store differently:

- A subnet that Active Directory considers part of a site may be connected over a slow link. In this situation, you probably want to configure another audit store and collectors that service audited computers in the remote subnet.
- A very large site may require multiple audit stores for load distribution. You can accomplish this by partitioning an Active Directory site into multiple audit stores based on subnets. Each subnet has its own audit store and set of collectors and audited computers.



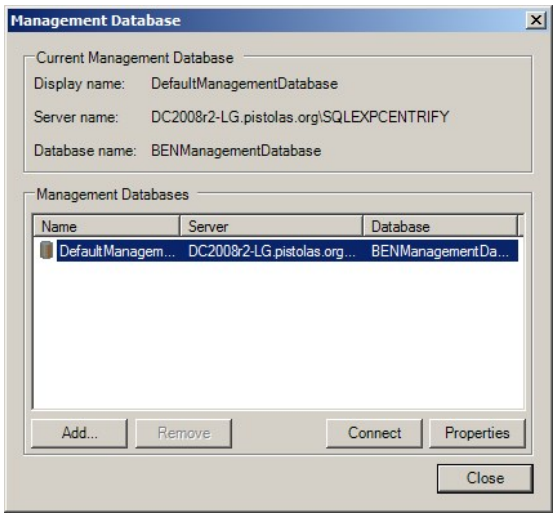
Two common audit store actions are:

- Adding a new audit store in a new site, and using the **Select Scope** page in the **Add Audit Store Wizard** to configure the site settings.
- Splitting an audit store in two, using the audit store's **Property** page to adjust the scope of the existing audit store, and then adding a new audit store.

To configure the audit store to support a particular subnet, click the **Subnet** radio button, and fill in the subnet IP address and mask.

Managing the Audit Management Database

The audit management database keeps track of where components are installed and information about the installation. To connect to the database or manage its properties, select a specific installation name in Audit Manager, right-click, then select **Management Databases**. From this dialog box, you can view information about the current audit management database, remove or connect to a management database, or change the properties for a management database.



Configuring Audit Management Database Scope

Select the audit management database you want to configure, then click **Properties**. From the Properties, click the **Scope** tab to configure audit management database scope. Click Add Site if you want to add a new Active Directory site to the management database or click Add Subnets to add a subnet for the management database to serve. Select the site or subnet from the list of sites or subnets found, then click **OK**. You can add or remove sites and subnets from the management database at any time using the Scope tab.

> **Note**: All components use Windows authentication whenever possible. However, an audit management database in another forest cannot connect to an audit store database using Windows authentication.

Setting Audit Management Database Security

Select the audit management database you want to configure, then click **Properties**. From the Properties, click the **Security** tab to configure security settings for the management database. Click the **Add** page to add groups or users to the list of trustees who can manage, modify, or remove installation-wide components. Type all or part of the user or group name, select the appropriate user or group from the results, then click **OK**.

Select the appropriate rights you want to grant to the selected Active Directory users or groups, and the operations that the users granted such rights ("trustees") are allowed to perform.

Full Control	All of the operations listed in the following rows of this table.
Change Permissions	Modify permissions on this audit management database.
Modify Name	Modify display name for this audit management database.
Manage Sites	Add a subnet or Active Directory site. Remove a subnet or Active Directory site.
Remove Database	Remove this audit management database from the installation.
Manage SQL Logins	Set the allowed incoming accounts for this audit management database. Set the outgoing account for this audit management database.

Manage Database Trace	Enable or disable database trace Export database trace

Only users with Change Permission permission on the audit management database can modify the user rights on the Security tab. By definition, the management database administrator has Full Control over all of the user rights and is an allowed incoming user.

Configuring the Maximum Memory for the Management Database

Because SQL Server dynamically acquires memory whenever it needs it until it reaches the maximum server memory you have configured, you should set constraints on how much physical memory it should be allowed to consume. You can use the formula described in Configuring the maximum memory for audit store databases to determine the maximum memory you should allow for the Microsoft SQL Server instances hosting the management database.

For more information about how to configure Microsoft SQL Server maximum memory setting and other memory options, see the following Microsoft article:

https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/server-memory-server-configuration-options

Removing an Audit Management Database

Select a specific installation name in Audit Manager, right-click, then select **Management Databases**. Select the audit management database you want to remove, then click **Remove**.

Because it is *not* recommended that you have multiple management databases in a single installation, you ordinarily would not separately remove an audit management database, but rather remove it as part of deleting an installation.

Maintaining Database Indexes

To ensure better performance and prevent database corruption, Centrify recommends you rebuild the database indexes for all the audit store databases and the management database as a regularly scheduled task that your run at least once a week. Rebuilding the indexes is especially important for the active audit store database to reduce fragmentation, but as a best practice you should rebuild indexes for all attached databases and the management database.

The following sample SQL statements illustrate how to rebuild all indexes on all the databases in one script:

```
=== BEGIN SQL statements ===
DECLARE @Database NVARCHAR(128)
DECLARE @Table NVARCHAR(128)
DECLARE @Command NVARCHAR(500)

-- To skip index rebuilding for a database, add its name to the list below
DECLARE DatabaseCursor CURSOR FOR
SELECT name FROM master.dbo.sysdatabases
WHERE name NOT IN ('master','msdb','tempdb','model')
ORDER BY 1

OPEN DatabaseCursor
FETCH NEXT FROM DatabaseCursor INTO @Database
WHILE @@FETCH_STATUS = 0
BEGIN
PRINT 'Processing database ' + @Database
SET @Command = 'DECLARE TableCursor CURSOR FOR SELECT
''['' + TABLE_CATALOG + '']."' + TABLE_SCHEMA

- ''].['' +
  TABLE_NAME + '']'' as TableName FROM [' + @Database
- '].INFORMATION_SCHEMA.TABLES
  WHERE TABLE_TYPE = ''BASE TABLE'''
  EXEC (@Command)
  OPEN TableCursor

FETCH NEXT FROM TableCursor INTO @Table
WHILE @@FETCH_STATUS = 0
BEGIN
PRINT 'Rebuilding all indexes on ' + @Table
SET @Command = 'ALTER INDEX ALL ON ' + @Table

- ' REBUILD'
  EXEC (@Command)
  FETCH NEXT FROM TableCursor INTO @Table
  END

CLOSE TableCursor
DEALLOCATE TableCursor

FETCH NEXT FROM DatabaseCursor INTO @Database
END
CLOSE DatabaseCursor
DEALLOCATE DatabaseCursor
=== END SQL statements ===
```

Managing Collectors

You can select the Collector node in Audit Manager to view details about each collector you have added to the installation. You can then expand the Collectors node and select an individual collector in the left pane to display information about the audited computers that send sessions to that collector in the right pane.

The following table describes the columns available in the right pane for collectors.

Collector	Name of the collector
IP Address	Location of the collector on the network
Status	Whether the collector is disconnected from or connected to the audit store. If a collector has never been successfully assigned to an audit store, it is not even shown in the left-pane list.
Uptime	How long a connected collector has been running since it was last booted
Last Update Time	The date and time of the last update received by the collector.
Port Number	The port through which the collector communicates with its assigned audited computers and audit store. Default is 5063.
Audit Store	The audit store to which this collector is assigned
Audit Store Database	The active database to which the collector is currently sending audit data
Connected Machines	The number of audited computers currently connected to this collector. Because agents can communicate with a collector only if the agents and collector are in the same Active Directory forest, this column only includes audited computers that are in the same forest as the collector.
Disconnected Machines	The number of audited computers of which the collector is aware but that are not currently connected to this collector. Note that the collector is only aware of audited computers that were at one time connected to it.
Collector Version	The version of the collector software installed on the computer.

Monitoring Collector Status

The Collector Control Panel is available from the Start menu on any Windows computer on which you have installed a collector.

The Collector Control Panel enables you to monitor the local collector by giving you an overview of collector connectivity and status, including the collector's current installation, audit store, audit store database, port number, and service status. To change the collector's port number, installation, or authentication, click **Configure**. If you change the collector configuration, it might take a minute for the change to be reflected in the Collector Control Panel.

You can also use the Collector Control Panel to start, stop, or restart the collector service, and to generate more detailed information about the status of the collector. To see detailed information about the installation, audit store, audit store database, trusted agents, and connectivity between components, click the **Troubleshooting** tab, then click **Diagnostics**. The collector will generate a report and display the information in a separate window.

Modifying the Command Prompt Recognized by the Collector

For the collector to identify the command events executed in a session, it must also be able to identify the command prompt. Although there are several characters that are commonly used and recognized by default, most computers also allow you to customize the command prompt. If a customized command prompt is not detected by the collector, commands will not be displayed properly in the session Events list, making it difficult for auditors to see the commands executed in a selected session.

To enable the collector to detect custom or unusual command prompts, you can add a registry key on the computer where the collector is installed and specify a text string or a regular expression that will match the command prompt.

To specify a regular expression for the command prompt:

1. Log on to the computer where the collector component is installed and running.
2. Open the Registry Editor.
3. Expand the **HKEY_LOCAL_MACHINE > SOFTWARE > Centrify > DirectAudit** registry.
4. Select the Collector component, right-click, then select **String Value**.
5. Type Prompt as the new key name.
6. Select the new Prompt key, right-click, then select **Modify**.
7. Type a text string or regular expression that will enable the collector to identify the command prompt you are using on computers you are auditing.

   If you don't define a registry value, the default regular expression ^[^#%›\$]*[#%›\$]\s* is used to detect the command prompt.

Removing Collectors

If you want to remove a collector, go to the installer and select the collector. The Collector Setup wizard Welcome page appears.

Because a collector is present on the computer, the next page enables you to select Change, Repair, or Remove the collector. Click **Remove**.

Managing Audited Computers and Agents

You can monitor agent status from the Audit Manager console. With audited computers selected in the left pane, Audit Manager displays the name and IP address for audited computers, whether the agent is currently connected or disconnected, and how long the agent has been running since last restarted. You can also see the collector to which the agent is sending data, the audit store and audit store database where the audit data is stored, and the version of the agent software installed on the computer.

Audited systems can be either a computer or a network device. Audit Manager displays two kinds of audited systems:

- **System-based**: A Windows or UNIX computer that is running an agent. You can access these systems either directly or from the Privileged Access Service Admin Portal.
- **Vault-based**: A Windows or UNIX computer or a network device that is not running an agent (agentless). You can access these systems from the Privileged Access Service Admin Portal.

Because agentless systems do not have an agent installed, the Audit Manager displays slightly different information for these kinds of systems. For these systems, you can see the name, IP address, the collector, audit store, and audit store database.

Monitoring Agent Status

You can use the dainfo -d command on audited Linux and UNIX computers to view information about the configuration, connectivity, and auditing status of the agent.

Configuring the UNIX Agent Off-line Database

If the UNIX agent is unable to connect to a collector, it spools the session data to local storage. When a collector becomes available, it then sends the spooled data to that collector.

By default, the minimum amount of allocated disk space that must be available to the offline database before spooling stops and warnings are posted to the agent error log is 10%. You can change this percentage by assigning a different value to spool.diskspace.min in the /etc/centrifyda/centrifyda.conf file. For example, to change the minimum to 15%, set the following value:

spool.diskspace.min: 15

If the threshold is reached and a collector is still not available, the agent stops spooling data, and further audit data is lost. If this happens frequently or unexpectedly, you may want to increase the disk space allocation.

Removing an Audited Computer

If an audited computer has been removed from the audit installation, the audited computer will continue to be listed on the Audit Manager as Disconnected. To remove the decommissioned audited computer, select Delete from its context menu.

Delegating Administrative Permissions

You can facilitate the administration of a large installation by delegating tasks and, if needed, setting up additional Audit Manager consoles.

Whoever creates the installation is the first administrator in the system, with full control of the entire installation and the ability to delegate administration tasks to any Active Directory user or group. You can grant permissions to other users on the Security tab of the Properties page for each component.

Publishing Installation Information

Audit Manager publishes information about your installation to a service connection point (SCP) object in Active Directory so that audited computers and collectors can look up the information. If the published locations for multiple SCPs in the same installation are not in synch, or if agents cannot read from at least one of the published locations, the agents are unable to determine which audit store is the best match for the sites and subnets, and so they do not attempt to connect to an audit store.

Permission to Publish to Active Directory

Only administrators who have been delegated permission to modify various attributes of the installation can publish those attributes to Active Directory.

If you do *not* have Active Directory permission to modify the installation, the updates are kept in the audit management database, and a message is issued to notify you that the installation information could not by updated in Active Directory.

Synchronizing Installation Information

If you have an Active Directory account with permission to modify the installation, you can click Synchronize the Installation Properties page Publication tab to publish the information.

In a multi-forest or DMZ environment, this tab lists multiple Active Directory locations to which to publish.

Managing Audit Roles

By default, each installation automatically has a Master Auditor role that has access to all audit data.The Master Auditor can read, replay, update review status, and delete all audit sessions in the installation. You cannot delete or change the permissions for the Master Auditor role itself. You can change the users or groups who are assigned to the Master Auditor role and the permissions granted to each role member, but you cannot make any other changes to this role. You can, however, create your own custom audit roles for the installation.

Creating Custom Audit Roles

Audit roles allow specific auditors to search and replay specific sessions, review specific events, or generate reports using the Audit Analyzer console based on the criteria you define. Each role specifies the criteria to use, the users and groups that are assigned to the role, and the specific permissions those users and groups have been granted.

For example, you might specify the criteria for filtering sessions to be only the session activity recorded on a particular audited computer or all UNIX sessions recorded after a specific date and time.

The collection of auditors is identified by specifying either explicit auditors, or an Active Directory group of auditors. Using Active Directory groups is recommended because this puts all of a user's privileges under the common Active Directory infrastructure.

For each audit role, you can also configure the specific permissions granted to each member of the role. For example, some audit roles might permit auditors to read and replay sessions but not update the status, add review comments, or delete the sessions to which they have access.

To create and assign audit roles:

1. Open Audit Manager and expand the audit installation to which you are connected.
2. Select Audit Roles, right-click, then select **Add Audit Role**.
3. Type a name and, optionally, a description of the audit role, then click **Next**.
4. Select the type of sessions—UNIX sessions, Windows sessions, or both UNIX and Windows sessions—to include for auditors assigned to this audit role, then click **Add** to specify filtering criteria for the role.
5. Select an attribute for filtering information from the list of Attributes.

   For example, you can match sessions based on the period of time in which they were active, based on a specific state, or based on Active Directory group membership. You can also match sessions based on the specific activity that took place during the session. For example, you can find sessions where specific UNIX commands or Windows applications were used.
6. Select the appropriate criteria for the attribute you have selected, then click **OK**.

   The specific selections you can make depend on the attribute selected. For example, if the attribute is **Review Status**, you can choose between "Equals" and "Not equals" and the specific review status you want to find, such as "To be Reviewed." If you select the attribute **Comment**, you can specify "Contains any of" and type the text string that you want to find any part of. If you select the attribute **Group**, you can select "Is (exactly)" and the user principal name (UPN) of an Active Directory group, such as adm-sf@acme.com.

   You can specify multiple attributes, by clicking Add and selecting additional attributes and criteria. You can test the filtering criteria you have added by clicking **Execute Query** and examining the results. When you have finished adding filters, click **Next**.
7. Select the privileges for the audit role, then click **Next**.
8. Review your settings for the audit role, click **Next**, then click **Finish**.

   You can assign users and groups to the audit role immediately by running the Assign users and Groups wizard or at a later time by rightclicking on the role name.
9. Type all or part of name to search for and select Active Directory users and groups to assign to the audit role.

Changing Audit Role Properties

After creating an audit role, you can modify its properties.

To change properties for an audit role:

1. Open Audit Manager and expand the audit installation to which you are connected.
2. Expand Audit Roles, select an audit role name and right-click, then select **Properties**.
3. Click the General tab to change the name or description of an audit role.
4. Click the Access tab to change the filtering attributes and criteria an audit role.
5. Click the Privilege tab to change what members of the audit role can do with the sessions matching the criteria you specify.
6. Click the Security tab to change permissions for the audit role itself.

   For example, you allow another user or group to change role membership for an audit role, you would click Security, then click Add to search for and select a user or group, the select the Change Role Membership permission to allow the selected user or group to modify the membership of the audit role.

Granting Permissions to Manage Audit Roles

Anyone you assign the Manage Audit Roles permission on an installation has full control over all of the audit roles for that installation. After you grant users or groups the Manage Audit Roles permission, they can create and remove roles, change the filtering criteria, modify audit role permissions for other users and group, and select the users or groups who are assigned to the role.

The following examples illustrate how users or groups granted the Manage Audit Roles permission might modify the audit roles for an installation:

- Assign the Master Auditor role to other users and groups.
- Create a UNIX Session Viewer role for UNIX auditors that allows them to view (read) UNIX sessions—but not replay, update, or delete—all UNIX sessions in the installation.
- Create a Finance Managers role that includes both UNIX and Windows sessions filtered by the Active Directory group Finance Operators, so that users assigned to the Finance Managers audit role can read, replay, update, and delete all of the session activity generated by members of the Finance Operators group, but no other groups.
- Create an audit role that enables investigators who are assigned to the role to read and replay only the activity captured when a specific command or application is used.

These are only a few examples of how you can use the Manage Audit Roles permission to define filtering criteria and privileges that control what different users or groups who are assigned to audit roles can see and do.

Querying and Reviewing Audited Activity

This section describes how to use Audit Analyzer to find and review the audited sessions and audit trail events in which you are interested. If you are the Master Auditor or been assigned an audit role, you can use Audit Analyzer to create and store queries that retrieve information from one or more audit stores. When you locate sessions or events of interest, you can review a summary of activity, play back all or part of the session, mark the session for follow-up, or change the status of the session.

The following topics are covered:

Delinea

Accessing Audited Sessions

Your access to audited sessions is controlled either through the audit roles you have been assigned, or through designation as a reviewer if you do not have an auditing role assigned to you. For more information on designating audit session reviewers, see Adding session reviewers without designating auditing roles.

If you have been assigned at least one audit role, or have been designated as a reviewer without an audit role, you can use Audit Analyzer to search for and replay the audited session activity collected from audited computers. Depending on the permissions defined for your audit role, you might also be able to annotate, update the status of, or delete the audited sessions to which you have access. If you have been designated as a reviewer of an audit session, you can only review and updated the status of the sessions to which you have access.

The first time you start Audit Analyzer, you are prompted to select an installation. If you have an audit role in that installation and the connection is successful, Audit Analyzer opens and displays the default categories for predefined queries:

- Audit Sessions
- Audit Events
- Reports

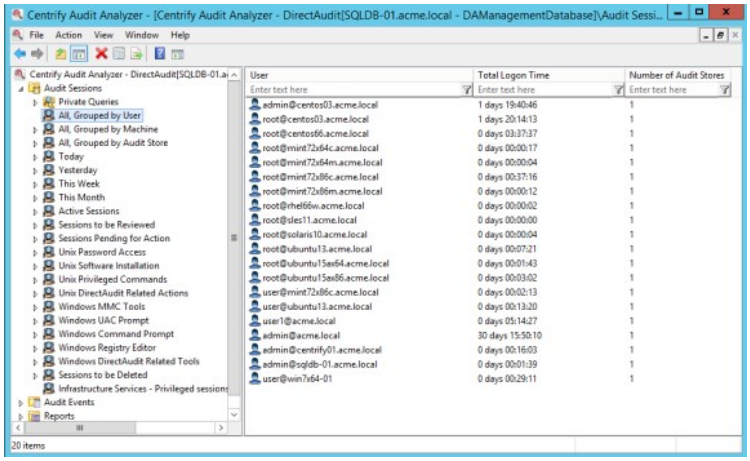
Predefined Queries for Audit Events

Audit Analyzer includes predefined queries that you can use to find the sessions that recorded audit trail events. To access the predefined queries for locating audit trail events, expand Audit Events. You can then select a predefined query to display a list of the audit trail events that meet the conditions of that query. You navigate to indexed lists of commands and events and replay sessions of interest for audit event queries in exactly the same way as audit session queries and you have the same options for viewing the activity captured. However, the details displayed for audit event queries are different from audit session queries.

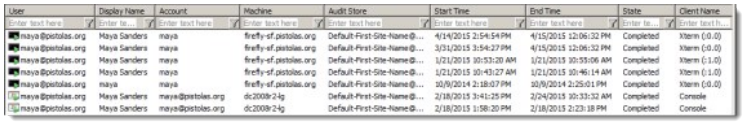
For each event, Audit Analyzer lists the name of the user, the name of the audited computer, the time of the event, the event name and description, and whether access was successful.

Predefined Queries for Audit Sessions

Audit Analyzer includes many predefined queries that you can use to find the sessions in which you are interested. To access the predefined queries, expand Audit Sessions. You can then select a predefined query to display a list of the audited sessions that meet the conditions of that query. For example, if you want to search for sessions by user, you can use the All, Grouped by User, then select the specific user whose sessions are of interest to see a list of all the sessions captured for that user. For example, in the right pane, you would select a user from the list:



After you select the user, Audit Analyzer displays detailed information about each of that user's sessions. For each session, Audit Analyzer lists the user name who started the session, the user display name, the account name used during the session, the name of the audited computer, the audit store where the session is stored, the start and end time for the session, current state, whether the audited session is a console or terminal client session, the review status of the session, any comments that have been added to the session, and the session size. For example:



Note that only completed sessions display the session size in Audit Analyzer.

Depending on the permissions associated with your audit role, you can right-click any session to view an indexed list of the activity captured, export the session activity to a comma-separated values file, update the review status for the session, or delete the session. If you have video capture auditing enabled for the installation, you can also select a session, right-click, then select **Replay** to review the session in the session player.

To view a description and definition for any predefined query, select the query, right-click, then select **Properties**. You can also export the query definition or the results from a query and perform other tasks on predefined queries. To perform any of these additional tasks, select the predefined query, right-click, then select the action you want to take.

Predefined Queries for Reports

Audit Analyzer includes predefined queries for generating reports. By default, the reports include information for all audited users, computers, and sessions. Select the type of report you are interested in generating, then specify additional criteria for filtering the report output. You can then save the modified report query or show the report.

If you click Show Report, the report is generated and displayed in a new window. You can then save the report as an HTML, PDF, CSV, or XML document.

User Activity Report

The default User Activity Report provides a detailed record of user actions for all audited users. The report includes the user name, the computer where the activity occurred, the time at which the activity occurred, and the event recorded. For example, if a user opened a Windows application or ran a UNIX command, the event would be recorded and included in the report you generate.

You should note that the User Activity Report does not include all desktop changes, such as navigation through directories using Windows Explorer. Instead, the report provides information about specific events. For example, the report will include information about when an application is opened, operations are performed, and when the application is closed. For more detailed information about user activity, you can enable video capture auditing for the installation and for specific desktops, applications, or commands using roles in Access Manager.

For information about enabling video capture auditing, see Enabling or disabling video capture auditing.

You can customize and filter the information included in a User Activity Report by specifying the query criteria and saving the report definition.

Privileged Activity Report

The default Privileged Activity Report provides a record of all actions taken with elevated privileges for all audited users and computers. The report includes the user name, the computer where the activity occurred, the time at which the activity occurred, and the event recorded. For example, if a user selected a role with administrative privileges, the event would be recorded and included in the report you generate.

You can customize and filter the information included in a Privileged Activity Report by specifying the query criteria and saving the report definition.

Centrify Zone Administration Activity Report

The default Centrify Zone Administration Activity Report provides a record of all zonerelated administrative actions taken for all audited users and computers. The report includes the user name, the computer where the activity occurred, the time at which the activity occurred, the client name, and the event recorded. For example, if an administrator created a new zone or delegated a management task to another user or group, the event would be recorded and included in the report you generate.

You can customize and filter the information included in a Centrify Zone Administration Activity Report by specifying the query criteria and saving the report definition.

Login by User Report

The default Login By User Report provides a record of both successful and failed login attempts for all audited users, computers, and sessions. The report includes the user name, the computer where the user attempted to log on, the time of the login attempt, and whether access was granted.

You can customize and filter the information included in a Login By User Report by specifying the query criteria and saving the report definition.

Login by Computer Report

The default Login By Computer Report provides a record of both successful and failed login attempts for all audited users, computers, and sessions. The report includes the user name, the computer where the user attempted to log on, the time of the login attempt, and whether access was granted.

You can customize and filter the information included in a Login By Computer Report by specifying the query criteria and saving the report definition.

Authorization Failure Report

The default Authorization Failure Report provides a record of authorization failure events for all audited users, computers, and sessions. The report includes the user name, the computer where the user attempted to log on or use a role, the time of the attempt, and the reason the user was denied access.

You can customize and filter the information included in a Authorization Failure Report by specifying the query criteria and saving the report definition.

Monitored Execution Report

If you have configured your auditing installation for advanced monitoring, then this Monitored Execution report shows the monitored commands being executed on the audited computers. This report includes information on commands that are run individually or as part of scripts. This report shows who ran one of the monitored commands even if that person is not an audited user.

The Monitored Execution report includes the user name, the computer where the commands were run, the time the command was run, the name of the command and the command arguments used, the process and parent process IDs, the "run as" user, the directory in which the command run, and whether the command was successful.

> **Note**: In the report, the Access Status column lists out whether the command was started successfully or not. This field does not describe whether the command completed successfully or not.
>
> **Note**: Advanced monitoring does not generate an audit trail event for commands for which you've enabled per-command auditing.

You can customize and filter the information included in a Monitored Execution report by specifying the query criteria and saving the report definition.

Detailed Execution Report

If you have configured your auditing installation to perform advanced monitoring, then this Detailed Execution report shows all of the commands being executed on the audited machines—including commands that are run as part of scripts or other commands.

The Detailed Execution report includes the user name, the computer where the activity occurred, the time at which the activity occurred, the command that was entered, the process and parent process IDs, the current directory, the actual command that was executed, the command arguments, the "run as" user, whether the command started or not (access status), and any additional access status details (such as "permission denied" if the access status is "failed").

> **Note**: In the report, the Access Status column lists out whether the command was started successfully or not. This field does not describe whether the command completed successfully or not.
>
> **Note**: Advanced monitoring does not generate an audit trail event for commands for which you've enabled per-command auditing.

You can customize and filter the information included in a Detailed Execution report by specifying the query criteria and saving the report definition.

File Monitor Report

If you have configured your auditing installation to perform advanced monitoring, the File Monitor report shows the sensitive files being modified by users on the audited machines. The File Monitor report includes any activity by any user (except root, -1) in the following protected areas on audited machines:

- /etc/
- /var/centrify/
- /var/centrifydc/
- /var/centrifyda/

> **Note**: The report includes the user name, the computer where the activity occurred, the time at which the activity occurred, the filename, the current directory, the kind of file access was attempted, if the file access was successful or not, the command that was used, the process and parent process IDs, and the "run as" user.
>
> If a monitored file is renamed, the report displays both the original and new filename. The order of filenames may differ slightly on each operating system.

MFA Failure Report

The default MFA Failure Report provides a record of multi-factor authentication (MFA) failure events for all audited users, computers, and sessions. The report includes the user's name, the computer where the user attempted to log on or use a role, the time of the attempt, and the reason that MFA authentication failed.

You can filter the information included in a MFA Failure Report by specifying the query criteria and saving the report definition.

Creating New Session Queries

You can create your own queries from existing queries or based on the criteria you define. Depending on the type of information you want to define as search criteria and whether you want to make the queries private or public, there are different type of queries you can define.

To search for audited sessions, you can create:

- Quick queries
- Private queries
- Shared queries

If you create a quick, private, or shared query, a new node is added to the Audit Analyzer console for that type of query under the Audit Sessions node. If you want to search for audit trail events, you can also create queries for audit events, which are added to Audit Analyzer under the Audit Events node.

Creating a new quick query

A quick query is a full-text search of the audit store database for a simple string or keyword. With a quick query, you can start typing the search string and see a list of potential matches from which you can select an item to look for sessions that contain the item. You should use quick queries when you want to find sessions based on a simple text string, such as a captured input or output, or based on a particular attributes, such as a user name or application, rather than using complex expressions.

To create a new quick query:

1. Open Audit Analyzer, select Audit Sessions, right-click, then select **New Quick Query**.
2. Type a search string into the search field.

   As you type, the Quick Query displays a list of possible matches that start with the text you are typing. For example, if you start typing the string "da" as the search term, the Quick Query list displays captured commands such as dacontrol, dad, and dadebug as potential matches:

The quick query uses SQL Server full text search.

The list of potential matches can include captured input and output, application names, user names, computer names, time stamps, and any other information stored in the audit store database.

   If a text string in the list is what you are looking for, select it. By default, the query will search for sessions that contain all of the text specified. If you want to search for any portion of the text specified, select **Find sessions containing ANY instead of ALL of the search terms**.

3. Click **Find** to display the matching logon sessions in the right pane.

Searching for a specific string

If you want to search for a specific string, you can enclose the command line string with quotation marks. For example, you can type "dacontrol i" to only return sessions that captured dacontrol with the -i option. If you type the same search string without quotation marks and select **Find sessions containing ANY instead of ALL of the search terms**, the quick query will return sessions that include dacontrol with and without the -i option.

Modifying a quick query

You can edit a quick query by selecting the query in the left pane, right-clicking, then selecting **Properties**. You can change the name and add a description on the **General** tab. Click the **Definition** tab to change the query text.

Creating a new private query

A private query is a set of search criteria that you define for your own use. Private queries are only visible to the auditor who creates them. You create private queries by selecting options in Audit Analyzer dialog boxes. Your selections are translated into complex expressions in the SQL Server query language. You can also save any predefined or shared query as a private query if you want to modify an existing query for private use.

To create a new private query:

1. Open Audit Analyzer, select **Audit Sessions**, right-click, then select **New Private Query**.
2. Type a name and description for the query.

   After you save the query, this information is available for viewing and editing on the General tab when you display the query's properties.
3. Select the type of sessions that you want the query to find.

   You can search for UNIX sessions, Windows sessions, and Linux Desktop sessions. By default, new queries search for all types of sessions.
4. Select an attribute for grouping query results, if applicable.

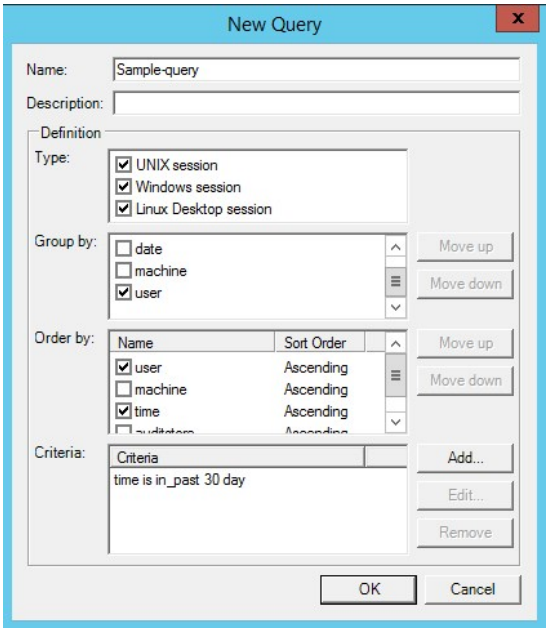
   You can select one or more attributes for grouping query results. If you specify more than one attribute, results are displayed as nested groups according to the order in which you specified the attributes. For example, if you select audit store, then user, then date, the query results are grouped by audit store, then by user for each audit store, then by date for each user.
5. Select an attribute for ordering query results within each group, if applicable.

   You can select ascending or descending sort order for each attribute. For example, you might group query results by user name and set the sort order for user to ascending, but the sort order for time to descending.
6. Click **Add** to add search criteria to filter the results of the query.
7. Select an appropriate attribute from the Attribute list based on the sessions you want to find.

   For example, you can search for sessions based on the period of time in which they were active or based on a specific state. You can also search for sessions based on the activity that took place during the session. For example, you can find sessions where specific UNIX commands or Windows applications were used.
8. Select the appropriate criteria for the attribute you have selected, then click **OK**.

   The specific selections you can make depend on the attribute selected. For example, if the attribute is **Review Status**, you can choose between "Equals" and "Not equals" and the specific review status you want to find., such as "To be Reviewed." If you select the attribute **Comment**, you can specify "Contains any of" and type the text string that you want to find any part of.

   When creating queries for user names or computers, you might want to use the "Starts with" option. If you use the default to match "Is (exactly)", you must include the fully qualified domain name of the user or computer.
9. Click **Add** to add another filter to the criteria for the query, or click **OK** to save the query and find the sessions that match the criteria you have specified.



Adding multiple filters to the query criteria

If you have more than one filter, different criteria attributes, such as Time and State, are separated by an implicit AND operation. Only sessions that match both criteria are returned. If you have repeated criteria attributes, for example, if you have two Time filters (time is not in past 10 days; time is in last month), the attributes are separated by an implicit OR operation. Sessions that match either criteria are returned.

Editing and removing filters from the query criteria

You can edit and remove any of the filters you specify. For example, if you are not finding the appropriate sessions, you might need to change or remove the criteria you have defined. After you have saved a query, you can right-click the query name, then select Properties to modify the query definition.

Specifying command or application filters in the query criteria

When you specify criteria for commands, applications, or outputs, the entry field displays a list of possible matches from audited sessions based on the text you are typing. For example, if you select "Windows Applications" as the attribute and "Contains any of" and start typing "word" as the text string, the entry field displays a list of possible matches that contain "word" in the application name. You can select a potential match or continue typing to specify the application by its display name or the executable file name. For example, you can specify winword.exe, Microsoft Word, or both.

Creating a New Shared Query

A shared query is a set of search criteria that you define for other auditors to use. Shared queries are visible to the auditors you specify. Only the auditor who creates a query can grant permission to other auditors to use the query. You create shared queries by selecting options in Audit Analyzer dialog boxes in exactly the same way as you create private queries. Your selections are then translated into complex expressions in the SQL Server query language. You can also convert a private or quick query to a shared query.

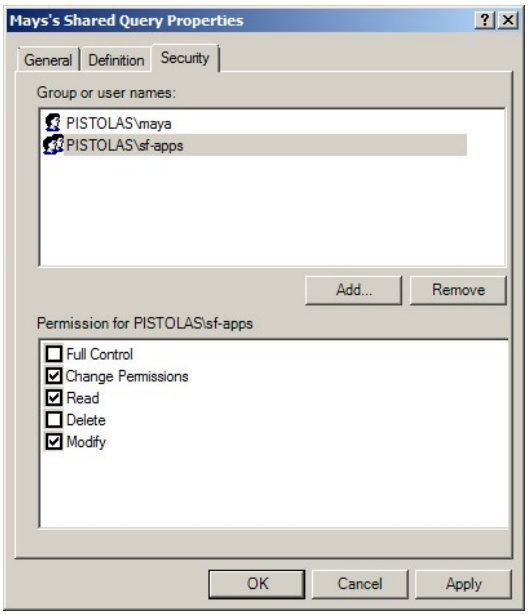
To create a new shared query:

1. Open Audit Analyzer, select Audit Sessions, right-click, then select **New Shared Query**.
2. Type the query name and select the session type, grouping, ordering, and other criteria for the query.

   If you need more information about specifying information for any field in the new query, press F1 to display context-sensitive help.
3. Click **Add** to add another filter to the criteria for the query, or click **OK** to save the query and find the sessions that match the criteria you have specified.
4. Expand **Shared Queries**, select the query name you specified in Step 2, right-click, then select **Properties**.

5. Click the **Security** tab.
6. Click **Add**.
7. Type the user or group name to identify the auditors who should have permission to use this query, then click **OK**.

   You can add multiple users or groups from the Select Users or Groups dialog box. You can also type part of the name, then click **Check Names** to look up user and group names.
8. Select each user or group, then select the appropriate permissions.



Searching for shared queries

After you publish queries and give other users permission to access them, other auditors can search for and select the shared queries they want to use. The shared queries are not automatically visible to users who have permission to use them.

To find shared queries you have permission to use:

1. Open Audit Analyzer, select Audit Sessions, right-click, then select **Open Shared Queries**.
2. Type the query name or click **Show existing queries**, then click **Find Now**.
3. Select one or more queries from the results returned, then click **OK** to add the query to your list of Shared Queries.

Creating queries for audit events

In addition to the predefined queries for audit events, you can create your own queries based on the criteria you define. Audit events are recorded for many activities, including both successful and failed operations. For example, you can search for events that are recorded when users attempt to log on and authentication fails or when users run commands or use applications with a role that grants elevated privileges. Audit trail events are also recorded when there are changes to the auditing infrastructure, and when there are changes to Centrify zones.

To specify the search criteria for a new audit event query:

1. Open Audit Analyzer, select Audit Events, right-click, then select **Query Audit Events**.
2. Type the query name and, optionally a description for the query.
3. Type a user name if you want to filter the event query by user name.

   You can specify one or more user names in userPrincipalName format (user@domain). Use semi-colons (;) to separate multiple user names. For example, to limit the search for audit events to events recorded for actions taken by the users ben, maya, and fred, you could type the following:

   ```
   ben;maya;fred
   ```
4. Type a computer name if you want to filter the event query by computer.

   You can specify multiple computer names separated by semi-colons.
5. Select the Event time option if you want to specify a time frame to filter the query based on when the event occurred.

   If you select this option, you can search for events that occurred:

   - before, not before, after, not after, between, or not between specific dates and times.
   - in or not in the last specified number of days, hours, or minutes.
   - during the specified period of time.
6. Select the Type option to search for events based on the type of activity performed.

   If you select this option, you must click **>** to view and select the event categories in which you are interested. For details about the type of events recorded in each category, select the category and review the Description displayed for that category.
7. Select the Result option to search for events based on the result of the activity performed.

   For example, you can use this option in combination with other options to search for only successful or failed operations.
8. Select the Role option, then a role name and zone if you want to filter the event query by role.
9. Select the Parameter option if you want to filter the query based on a specific parameter.

   If you select this option, you must click **>** to view and select the event parameters that are currently available and in which you are interested.
10. Click **OK** to save and run the new query.

After you create a new query, you can export the query definition or its results, email it to others, or modify its properties.

How Access Manager Roles Affect Audit Trail Events

If you only enable auditing without access control and privilege management features, audit trail events are recorded for all successful and failed operations on audited computers. The events are stored in the audit store database and can be returned in response to queries. These events are not associated with roles, so you should not use the Role filter in your query definition.

If you enable auditing with access control and privilege management, however, user activity is only recorded when a role with “auditing required” or “audit if possible” setting is used to perform one or more tasks. In most cases, roles that allow users to perform tasks using elevated privileges or in a restricted shell environment are configured with one of these audit settings. By default, the Windows Login and UNIX Login roles are also configured to “audit if possible” to capture all audit trail events on the computers where the auditing service is running. If a role is configured with audit not requested or required, only audit trail events are recorded.

If the auditing service is running on the computer where the user logs on or where the administrative tasks are performed, the audit trail event is collected and

transferred to the audit store database. Only the audit trail events that are captured and stored in the audit store database can be returned in response to audit event queries. Therefore, from Audit Analyzer, you can only query and report on audit trail events that are stored in the audit store database while a user performs tasks in an audited role on an audited computer.

Querying by Audit Event Type or by Role

In many cases, querying for audit trail events by event type produces more predictable results than querying for events by role. For example, to query for successful and failed login attempts, select Type, then select the Login Event category. In this particular case, the Windows Login and UNIX Login roles do not—as a user's effective role—capture successful and failed login attempts, so they should not be used as filters for querying successful and failed login events.

If you query using the Role filter, Audit Analyzer only returns the audit trail events associated with the selected role. In some cases, this might be the information you are looking for—for example, to review the execution of commands using a role with elevated privileges. On UNIX computers, however, many audit trail events are not linked directly to the actions taken with a specific role. For example, on a Linux or UNIX computer with the auditing service running, many command-line activities record audit trail events. These events are stored in the audit store database and can be queried, but are not associated with any role and not reported if you select a role filter.

Populating and Deleting the Roles Available

The list of roles available for querying is based on the roles you have defined using Access Manager. If you add a role definition, the new role displays in the list of roles when an audit trail about the role is generated.

If you delete a role from all zones, however, it will remain in the list until the last session that has events associated with that role is deleted or the audit store database is detached.

Organizing queries in custom folders

By default, queries are organized into folders by type. You can choose to organize your queries in other ways. For example, you can create a custom hierarchy of folders and move your queries into those folders. The folder information is stored locally and does not affect other auditors, so each auditor can have a private folder structure for favorite queries.

To create a custom folder hierarchy:

1. Open Audit Analyzer, select Audit Sessions, right-click, then select **New Folder**.
2. Select the new folder, right-click, then select **Rename** and type a new folder name.
3. Right-click the new top-level folder, then select **New Folder** to create sub-folders.

Exporting and Importing Query Definitions

You can export and import query definitions from one Audit Analyzer console to another to make queries available to different groups of auditors. You can also export query definitions for individual queries or for queries stored a custom folder hierarchy. For example, if you have a custom "Queries Required at All Sites" folder, you can select that folder and only export those query definitions.

To export query definitions:

1. Open Audit Analyzer, select the Audit Analyzer root node, right-click, then select **Export Query Definitions**.
2. Select a location and type a file name, then click **Save**.

   All of the query definitions are saved to an xml file.

To import query definitions:

1. Open Audit Analyzer, select the Audit Analyzer root node, right-click, then select **Import Query Definitions**.
2. Navigate to the location that contains the .xml file you want to import, then click **Open**.

   The imported queries are created as private queries. If you have an audit role with Manage Shared Query privileges, you can publish the imported queries as shared queries.

Displaying Session Information

After you select a query to see a list of sessions, such as the **Today** query to see a list of today's sessions or an individual user to see a list of sessions for that user, you can view an indexed list of the activity that took place during any of the individual UNIX or Windows sessions captured.

For example, you can select a Windows session, right-click, then select **Indexed Event List** to review a list of the applications that were opened during the session, in the order in which they were opened, the title of the active window, the type of activity, the desktop role used to access the application, and whether audit data was captured for the role being used. If you have video capture auditing enabled for the installation, you can replay the session entirely or from any point in the indexed list.

Similarly, for UNIX sessions, you can select a specific session, right-click, then select **Indexed Command List** to display a list of commands executed and the order they occurred. If you have video capture auditing enabled for the installation, you can replay the session entirely or from any point in the indexed list.

Adding Session Reviewers without Designating Auditing Roles

If you have been assigned an auditing role that allows you to replay, delete, and update the status of an auditing session, you can also designate users or groups the permission to replay and update the status of that session, even if they do not have an assigned auditing role.

> **Note**: Users and groups assigned as session reviewers cannot delete auditing sessions, and therefore cannot change the reviewer list for the sessions available to them.

To designate users or groups as reviewers of one or more auditing sessions:

1. In Audit Analyzer, select the session or sessions you want to be reviewed. You can do this by selecting the predefined groupings or by defining specific criteria using a query.
2. Right-click the selected sessions and select **Set Reviewers**.
3. Type all or part of the name of the user or group that you want to add to the list of reviewers and click **Check Names**.

   If you would like to add multiple reviewers, separate the full or partial names by semicolons.
4. Click **OK**.

To remove reviewers from a session, right-click the session and select **Clear Reviewers**.

Changing the Review Status for Audited Sessions

You can use the review status to keep track of audited sessions. For example, if you have a formal review process, you can change the state of sessions to indicate whether they are in the queue to be reviewed, have been reviewed, are awaiting some type of action, or should be deleted. For each change of state, you can add comments to more fully document what's been done or if any follow-up by another auditor is required.

By default, all audited sessions start with a review status of **None**.

To update the review status for a session:

1. Open Audit Analyzer and navigate to the session.
2. Select the session in the right pane, right-click, then select **Update Review Status**.
3. Select the appropriate new status.

   For example, select **To be Reviewed** if the session requires a review or **To be Deleted** if the session has no activity requiring further review or data that must be retained.
4. Type any notes for yourself or other auditors in the Comments dialog box, then click **OK**.

Viewing status history

The changes you and other auditors make to the review status for a session are recorded and cumulative, so that you can view the complete status change history for any session.

To view the status change history for a session:

1. Open Audit Analyzer and navigate to the session.
2. Select the session in the right pane, right-click, then select **Properties**.
3. Click the **Review Status** tab.

   Changes to the review status are listed with the most recent change at the top of the list and proceeding back in historical order. You can select any review status change in the list to see who made the change and any comments recorded when the change was made.

Adding comments to a session

The comments associated with a session are cumulative. For example, if you select **To Be Reviewed** and type a comment, then later change the state to **Reviewed** with another comment, both comments are displayed on the **Comments** tab if you view the session's Properties.

You can also add comments to a session without changing its review status. To add comments to a session without changing the review status, right-click the session, select **Properties**, then click the **Comments** tab. You can use this tab to record detailed information about sessions of interest. You can also use the Review Status attribute to find sessions by review status, and the Comment attribute to find sessions by comment text.

Reviewing and deleting your own sessions

By default, you can update the review status, add comments, and delete your own sessions if you have an audit role with the appropriate permissions. However, there are installation wide options to prevent any users from updating the review status or deleting their own sessions. These installation-wide options take precedence over your audit role permissions. Depending on how these options are set, you might be prevented from updating the review status and adding comments to your own sessions, prevented from deleting any of your own sessions, or prevented from both. If either installation-wide option is set, you might be blocked when you attempt to add a comment or delete a session.

Playing Back a Session

If you select the **Enable video capture auditing option** for an installation, you can replay session activity captured on audited Windows or UNIX computers.

For Windows computers, the video record captures desktop activity when users select roles with auditing enabled.

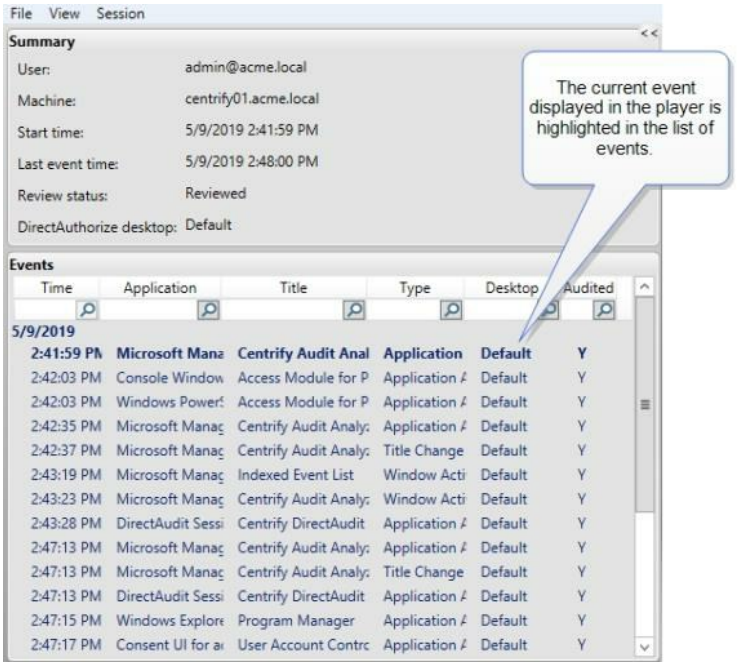
For UNIX sessions, the video record captures complete input and output typed in a UNIX shell during a session.

If the Replay option is available for a session on the rightclick menu, you can view a summary of the commands executed or applications opened in the session player. You can also search for commands, parameters, or events, control the playback speed and magnification from the session player, and update the session review status.

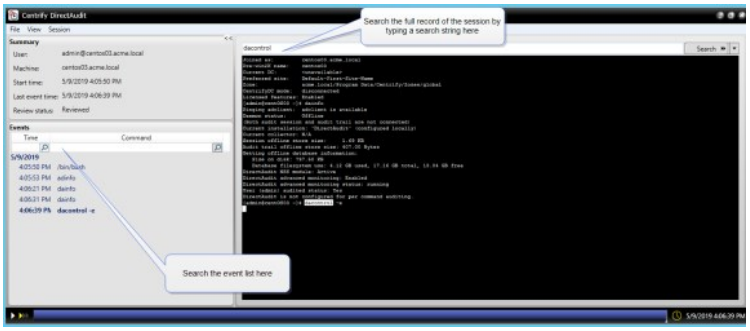
To play back a session when video capture auditing is enabled:

1. Open Audit Analyzer and navigate to the session.
2. Select the session in the right pane, right-click, then select **Replay** to open the session player.

   The left pane of the session player displays a summary of activity similar to the indexed list. For example, if the session is a Windows session:



   You can search on any column to find events of interest. If the session is a UNIX session, you can search the full session for any text string. For example, if you are playing a UNIX session, the right pane displays the shell session and a search field.



3. Click the **Play/Pause** icon ( ) at the bottom of the session player to start or stop the session you are viewing.

   You can also fast forward session playback by clicking the **Speed control** icon to play back at 2x or 3x the normal speed. The dark blue playback line across the bottom of the window represents the total time of the session. You can drag the **Timepoint needle** to go directly to a specific point in the session.

   The **Real-time** icon toggles to allow you to play back a session as it was recorded in real time or move swiftly from one user action to the next. The **Session point** in the lower right corner identifies the date and time of the current point in the session playback.

4. To update the session review status:

   1. Select **Session** › **Update Review Status**, and then select the desired review status.
   2. Add your review comments and click **OK**.

      The updated session review status displays in the session player.

5. Close the session player.

Starting the Session Player Separately

In most cases, you start the session player from Audit Analyzer. However, you can also start the session player from a Windows command prompt using standard Windows command line options or by specifying a Uniform Resource Identifier (URI).

Using Window command line options

If you use the Windows command line to start the session play, the installation name and session ID are required. The other arguments are optional.

daplayer /installation=*installation_name* /id=*session_guid*
[/conn=*auditserver_connection_string*]
[/store=*auditstore_ID*]
[/time=*timestamp*]

For example:

daplayer.exe /installation=MyInstallation
/id=""
/store=1

If you don't specify the audit server connection string, the session player attempts to bind to an appropriate audit management database. The session player can replay sessions from only one audit store, but the audit store ID is optional because sessions usually reside in a single audit store. An individual session can span multiple audit store databases within a single audit store. If a session spans multiple audit stores, that is, different subnets or sites, you should specify which audit store to play it from.

The timestamp option is a 32-bit integer that tells the session player to jump to the point where the event of interest occurred.

Using the Uniform Resource Identifier (URI)

The Uniform Resource Identifier identifies the session player, the installation name, and the session GUID for each session. This format is especially useful when used with the **Copy Session URI** menu item. The URI link can then be pasted into an email or instant messenger message. On a computer where Audit Analyzer is installed, the recipient can simply click on the URI link and the session player starts automatically.

Playing Back a Session from a Web Browser

On computers that have Audit Analyzer installed, you can also play back sessions from a web browser. Because the cda:// protocol is automatically registered on the computer with Audit Analyzer, you can use a web browser to replay a specific session. If you want to play back a session from a web browser, you can extract the installation and session identifier from the session URI.

To get the installation and session identifier:

1. Select a session and right-click or open the session in the session player, then select **File > Copy Session URI**.

2. Open a text editor and paste the session URI into the file.
3. Delete the portion of the URI that identifies the player, so that only the installation and the object GUID remain.

   For example, if the URI looks like this:

   rep://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395

   Remove the first part of the URI so that you only have the installation name and session identifier:

   //myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395

To play back a specific session from a web browser:

1. Open a web browser.
2. Type the installation name and session ID in the address bar of the web browser:

   cda://‹*installationName*›/‹*session_id*›

   For example:

   cda://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395

   The session player opens and plays the specified session.

Exporting Sessions

Depending on whether you have selected the **Enable video capture auditing option** for an installation, you might have different options for exporting session data to a file. The options available also depend on whether the session activity was captured on an audited Windows computer or an audited UNIX computer.

To view your export options, select the session and right-click or open the session in the session player, then click the File menu. Depending on the session type or installation settings, you might see the following export options:

Export to Command List

Exports the time stamp and UNIX shell commands as comma separated values (csv) in a text file. The file contains the same information as displayed in the Indexed Command List for UNIX sessions.

Export to Event List

Exports the time stamp, application name, and other details as separated values (csv) in a text file. The file contains the same information as displayed in the Indexed Event List for Windows sessions.

Copy Session URI

Copies the URI of the selected session to the clipboard. You can then paste the URI into a web browser to open the session.

Check Session Data Integrity

Checks the session for any possible data tampering. If the session is fine, a message displays that the data integrity check passed. If the session has been tampered with, a message displays with details of what data was affected.

Export to TXT

Saves the selected UNIX session(s) or UNIX session(s) and user input (stdin) as a plain text file.

If you selected multiple sessions, a message displays that asks you if you want to export the multiple sessions to a single file. Click **Yes** to save the sessions in a single file or **No** to save the sessions in separate files.

If you select the Export Session with User Inputs option, user input is noted with a line number of K or "keyboard" input.

Export Detailed Executions

Saves the session in HTML, PDF, CSV, or XML format if you have enabled advanced monitoring and the session includes any detailed executions.

Export to CDF

Saves the selected Windows session in Computable Document Format. You can then open the CDF file with the session player (daplayer *filename*.cdf). Because the session player reads the session information directly from the CDF file, you don't need to specify an installation name or connect to a database to replay the session.

Export to WMV

Saves the selected session in Microsoft Windows Media Video format. You can use Windows Media Player or other media players to play back sessions in this format. However, sessions exported to WMV files do not include the summary information such as the user name, the computer name, start and end times, or the list of events captured.

Deleting Sessions

Auditing allows you collect detailed information about activity in your organization. In some cases, however, you might have sessions that collect information that you are not interested in capturing or include information that you don't want to store or make available to other auditors. For example, you might find there are sessions with very little activity or sessions that have been reviewed and are no longer needed. You might also notice that there are sessions that captured personally-identifying or medical data that other auditors should not be allowed to see. To handle these cases, you can selectively delete sessions from the audit store database.

In most cases, if you are the Master Auditor or have been granted permission to change the status of a session, you can mark sessions for deletion in Audit Analyzer. As noted in Reviewing and deleting your own sessions, however, you might be prevented from deleting your own sessions if the installation-level setting prevents users from deleting their own sessions.

To delete a specific session:

1. Open Audit Analyzer console, then use a predefined or custom query to find the sessions that you want to delete.
2. Select the sessions that you want to delete.
3. Right-click, then select **Delete**.

   Audit Analyzer displays a confirmation message indicating that the deletion cannot be reversed.
4. Click **Yes** to continue.

To delete all sessions in a query:

1. Open Audit Analyzer, right-click a query node, then select **Delete All Sessions**.

   Audit Analyzer prompts you to confirm the deletion of all sessions returned by the query.
2. Click **Yes** to continue.

   Audit Analyzer prompts you to confirm the deletion of sessions with a review status of To be Reviewed or Pending for Action.
3. Click **Yes** to delete those sessions, or click **No** to continue the deletion of other sessions but preserve the sessions marked for retention.

   While the delete operations runs, you can click Stop Delete if needed. Sessions are partially deleted up until the point where the delete operation was cancelled.

Viewing Sessions Outside of Audit Analyzer

You can view audited sessions while working in other Centrify management consoles. For example, on computers that have Audit Analyzer and Access Manager installed, you can start the session player from Access Manager or from Active Directory Users and Computers. You can also launch the session player by itself or from a web page or a software program.

Viewing Sessions from Access Manager

On computers where both the Access Manager console and the Audit Analyzer console are installed, you can search for and view sessions directly from the Access Manager console.

To view audited sessions in Access Manager:

1. Navigate to a computer, user, or role assignments node in the left pane of Access Manager.
2. In the right pane, right-click the object and select **View DirectAudit Sessions**.
3. Specify any additional criteria, then click **Find**.

Viewing Sessions in Active Directory Users and Computers

On computers where you have Active Directory Users and Computers with Access Manager properties and Audit Analyzer, you can view audited sessions directly from Active Directory Users and Computers.

To view audited sessions from Active Directory Users and Computers:

1. Navigate to the Users node in the left pane of the Active Directory Users and Computers.
2. In the right pane, right-click the user and select **All Tasks > View DirectAuditSessions**.
3. Specify any additional criteria, then click **Find**.

Using Find Sessions

Find Sessions is a separate executable file, installed in the same directory as Audit Analyzer, that you can use to find and open audited sessions. The program provides a graphical user interface and a command line interface for specifying the search criteria. You can use either interface to find sessions of interest. From the Find Sessions graphical user interface, you can also replay, update the review status, view the desktops used for any sessions found, display the list of indexed commands or events, and copy the session URI.

To start Find Sessions from the Windows command line, you can type the following in a Command prompt window:

findsessions /ia

Specifying the Sessions to Find

You can use the Common or Advanced search criteria to find sessions of interest. The Find Sessions dialog box then displays the results that match the criteria you specify. In most cases, you can find the sessions you are interested in through some combination of user name, computer name, and session time displayed on the Common tab. If you want to specify additional criteria, such as review status or auditor name, you can click the Advanced tab.

Using the Command Line Interface

You can run Find Sessions as a command line utility on computers where Audit Analyzer is installed. The command line interface can be useful, for example, if you may want to find, export, or delete sessions as part of a script. You can view usage information for the command line interface using the /help option. Specify search criteria for finding sessions using the following format:

findsessions /i="*InstallationName*" /u="*username*" /m="*computerName*" /t="*yyyyMM-dd*"

Using a web browser to access sessions

On computers that have Audit Analyzer installed, you can also find and play back sessions from a web browser. Because the cda:// protocol is automatically registered on the computer with Audit Analyzer, you can use a web browser to open Find Sessions or to replay a specific session. For example, you can embed a cda:// link in a web page to automatically generate a list of sessions, or you might want to embed a link to a session or set of sessions in a web-based report or event notification.

You must be able to specify a query using AQL syntax to open Find Sessions from a web browser. If you want to start playing back a session from a web

browser, you must know the session identifier. You can extract the session identifier from the session URI.

To start Find Sessions from a web browser:

1. Open a web browser.
2. Type the installation name and a search string using AQL syntax in the address bar of the web browser.

   For example, if you want to search an installation named MyInstallation5 for sessions that involved the Administrator user, you would type the following in the address bar:

   cda://DefaultInstallation5/?search=\"1 user=\"Administrator*\"\"
3. Click **Allow** to open the Find Sessions with the Advanced tab displayed and "user=Administrator*" listed for the Define Criteria.
4. Click **Find Now** to find sessions matching the criteria you specified.

For more information about using Find Sessions, see the Find Sessions help.

Using Tags with Sessions

Tags can be a helpful way to quickly find sessions that meet particular criteria, such as sessions that contain PII (Personally Identifiable Information). You can add one or more keywords to sessions and then use those keywords in your session queries to find those specific audited sessions.

Assigning Tags to Sessions

You can assign tags to sessions or remove tags from sessions if you're assigned to an audit role with "Update Status" permission. To apply multiple tags, separate each tag by a space.

You can use assign one or more keywords to sessions in the following ways:

- In the Audit Analyzer query results pane, right-click a session > select **Add Tags**.

  In this way, you tag the session itself and not a specific point in time of the session.

- In a session's Indexed Command List, select a command and select **Add Tags**.

  When adding a tag this way, the tag is associated with a specific timestamp in the session.

- While replaying a session, select **Session** > **Add Tags**.

  When adding a tag this way, the tag is associated with a specific timestamp in the session if you have paused the session. If you are playing the session when you add a tag, the tag isn't associated with a timestamp.

- Use the PowerShell cmdlets to tag sessions. For details, please see the PowerShell command help for the following cmdlets:

  - New-CdaAuditSessionTag
  - Get-CdaAuditSessionTag
  - Remove-CdaAuditSessionTag
  - Get-CdaAuditSession

When you add tags, each tag is a single word; to add multiple tags, just separate them with a space. Each tag needs to be at least 3 characters long.

Viewing Tags Associated with a Session

You can view tags that are associated with a session if you're assigned to an audit role with "Read" permission. You can also remove the tag after reviewing the session, if you also have the "Update Status" permission.

To view tags associated with a session

1. In Audit Analyzer, navigate to a list of sessions.
2. Right-click a session and select **Properties**.

   The Session Properties dialog box opens.
3. Click the **Tags** tab.

   The Tags tab lists all tags and related information that are associated with the selected session. In addition to tags, you can also see who added the tag to the session and when they added the tag.
4. If desired, you can click **Remove** to remove the associated tag or **Replay** to replay the session.
5. Click **OK** to save your changes and return to the Audit Analyzer window.

Searching for Sessions Associated with Tags

You can enter tags as search criteria for either quick queries, private, or shared queries. For a quick query, enter the tag name as the search term.

For a private or shared query, when you add criteria for the query you can select Tag as an attribute to use in the query.

Advanced Monitoring

The Centrify Audit & Monitoring Service captures input and output for audited users and commands and then uses this information to provide a history of executed commands.

However, you may want to gather additional information about which users and what programs are accessing or modifying production systems. For example, you may want to know when any user runs a highly privileged program, even if the user runs it from a script or by modifying system configuration files. You can use advanced monitoring to capture these kinds of activities.

One of the big differences in advanced monitoring is that you can track when *any* user performs a particular activity, not just an audited user.

Advanced monitoring uses the Linux system auditing tools to capture the following user and program activity:

When *any* user executes a particular program, not just audited users.	Audit Analyzer Linux agent syslog Monitored Execution report Monitored Execution List	yes
When *any* user (not just audited users) attempts to modify system configuration files in monitored directories specified by an administrator.	Audit Analyzer Linux agent syslog File Monitor report	yes
Which programs are executed in an audited session, *regardless of how the program is invoked*-- whether it's run by way of a script, the use of a command alias, and so forth.	Audit Analyzer Detailed Execution report	no - there would be too many events for the information to be useful.

Set up Advanced Monitoring

To configure advanced monitoring, make sure that your computer meets the requirements, make some configuration changes in the centrifyda.conf file, and then enable advanced monitoring either by using the dacontrol command or the "Enable Advanced Monitoring" group policy.

Advanced Monitoring Requirements

- Currently, Centrify supports only 64-bit Linux distributions from RedHat (RHEL, Fedora, CentOS). For more information about supported platforms and versions, please refer to the current Audit & Monitoring Service release notes.
- Verify that you have the Linux audit package running. For example, run this command:

  ```
  rpm -qa audit
  ```

- Ensure that the Linux audit package that you have is supported for use with Centrify Audit & Monitoring Service. Version 1.2.8 or later of Linux audit package is required. However, the Audit & Monitoring Service prefers the Linux audit package version 2.4.5 or later because earlier versions may have issues with startup.
- Ensure that your collector and audit store database are running Server Suite 2017 or 2017.1, or Infrastructure Services 2017.2 or later.

Configuring Advanced Monitoring

You have some options and choices as to how you configure advanced monitoring. To use any of these parameters, you must also enable advanced monitoring (by using the dareload -m command or the "Enable Advanced Monitoring" group policy). Here's a list of the configuration parameters that you can edit in the centrifyda.conf file:

- **event.file.monitor**

  Use the event.file.monitor parameter to enable advanced monitoring for configuration files.

- **event.file.monitor.process.skiplist**

  For any areas that you've specified to monitor (using event.file.monitor), use the event.file.monitor.process.skiplistparameter to ignore any specific

processes in those areas.

- **event.file.monitor.user.skiplist**

  Use the event.file.monitor.user.skiplist parameter to specify a list of users to exclude from advanced monitoring for files. For these users, the auditing service does not record any write access to directories specified in event.file.monitor.

- **event.execution.monitor**

  Use the event.execution.monitor parameter to monitor all programs that users run in an audited session.

- **event.monitor.commands**

  Use the event.monitor.commands parameter to specify a list of commands to monitor. Be sure to list each command using the full path name of the command. The auditing service generates an audit trail event when a user runs any of these monitored commands, unless the user is listed in the event.monitor.commands.user.skiplist parameter.

- **event.monitor.commands.user.skiplist**

  Use the event.execution.monitor.user.skiplist parameter to specify a list of users to exclude from advanced monitoring for program execution. For these users, the auditing service does not record any programs that they run, even when the parameter event.execution.monitor is set to true.

After you make the configuration changes in the centrifyda.conf file, run the dareload -m command to apply the changes.

Enabling Advanced Monitoring

After you've made your configuration changes in the centrifyda.conf file, the next step is to enable advanced monitoring.

To enable advanced monitoring:

- Run the following command:

  dacontrol -m

- Or, use the Enable Advanced Monitoring group policy.

To disable advanced monitoring:

- Run the following command:

  dacontrol -n

- Or, discontinue using the Enable Advanced Monitoring group policy.

Using the Advanced Monitoring Reports

These reports provide details on what your advanced monitoring configuration has tracked:

- **Monitored execution report**

  If you have configured your auditing installation for advanced monitoring, then this Monitored Execution Report provides a detailed record of the sessions where a user ran one of the commands that you've configured to monitor. This report shows who ran one of the monitored commands even if that person is not an audited user. Also, this report includes information on commands that are run individually or as part of scripts.

- **Detailed execution report**

  If you have configured your auditing installation to perform advanced monitoring, then this Detailed Execution report shows all of the commands being executed on the audited machines—including commands that are run as part of scripts or other commands.

- **File monitor report**

  The File Monitor report shows the sensitive files being modified by users on the audited machines. The File Monitor report includes any activity by any user (except root) in the following protected areas on audited computers:

  - /etc/

- /var/centrifydc/
- /var/centrifyda/
- /var/centrify/

Troubleshooting and Common Questions

This chapter describes how to view and manage log files and diagnostics for components of the auditing infrastructure on UNIX computers. This chapter also describes how to identify and resolve common problems you might encounter when auditing user activity or managing the auditing infrastructure.

Checking the Status of the UNIX Agent

After you install and enable auditing for a UNIX computer, you can check the status of the agent using the dainfo command to verify the connection to the correct installation. For example, the agent might not automatically connect to the installation if you use an installation name other than DefaultInstallation.

To check the status of the agent and the auditing infrastructure, run the following command as a user with root privileges:

```
dainfo --diag
```

The `--diag` option returns detailed information about the local computer and about the installations, audit stores, trusted collectors, trusted agents, and the active audit store database that the agent is sending its data to. The diagnostic output also includes details about the Active Directory location and object identifier for each installation.

Configuring the Installation for an Agent

If the command indicates that the status is offline or the installation is not configured, use dacontrol to explicitly identify the correct installation. For example:

```
dacontrol -i installation_name
```

You can then rerun `dainfo --diag` to verify the installation is configured correctly. Note that you cannot use `dacontrol` to connect to a different installation name if the installation is configured using the Installation group policy. In a secure installation, the Installation group policy identifies the Active Directory location that contains the service connection point object for the installation. If you are not using group policy to identify the installation, you can manually configure agents and collectors to use a specific installation name.

Checking for Disconnected Agents using Audit Manager

You can also use Audit Manager to see the status of all agents in the installation. I any agent is listed as Disconnected, you should check whether the audited computer is shut down, If the audited computer is not shut down, the agent might be outside the scope of any audit store or unable to find a collector. Use the diagnostic services to check communication between components.

Starting and Stopping the UNIX Agent

In most cases, the UNIX agent is automatically started when an audited computer is first powered on and remains running until the audited computer is shut down. Starting the agent when a computer starts up ensures the agent can capture activity for all shell sessions.

Although you typically start and stop the dad process as part of a computer's startup and shutdown scripts, you can also start the agent directly from the command line on a local computer.

If the agent is not running, run the following command to start it:

```
/usr/share/centrifydc/bin/centrifyda start
```

Detecting the Server Suite Installation Status

If you're encountering any issues with your Server Suite installation, you can run the `dacheck` program on your UNIX computers. The `dacheck` command detects the following errors in your Server Suite installation:

- Auditing binaries linkage problems
- Disk space
- DNS, collector, dad, adclient health
- Logging status
- Auditing file permissions/ownership
- Auditing installation configuration
- If ActiveDirectory joined
- Auditing database integrity
- If root in user.ignore and other criteria that affect root login

- var/centrifyda, /tmp write permission
- nsswitch.conf (or method.cfg, user.cfg for AIX)
- Selinux status
- Nscd (pwgrd) status
- User's cdax/real shell existence, permission, ownership.
- DNS Reverse lookup for collector's hostname
- Report Domain Controller

To check the status of the agent and the auditing infrastructure, run the following command as a user with root privileges:

- dacheck

The dacheck command is available in the same location as the adcheck command: /usr/share/centrifydc/bin.

Viewing and Changing Log File Settings

Log files are text files that record information about operations performed by auditing components on a local computer. If you have administrative privileges on a computer, you can open log files with any text editor.

You can view log files, change the location of the log file, and change the level of detail recorded in the log file from the Log Settings dialog box. Depending on the computer you are using, there are different ways to open the Log Settings.

Enabling Detailed Logging for Linux and UNIX Computers

In most cases, troubleshooting auditing-related issues requires information about the operation of the agent, the collector service, and database activity. For performance reasons, you should only enable agent logging when you need to capture detailed information about agent operations. For troubleshooting purposes, however, you can use the dadebug command to turn on detailed logging.

To enable audit-related logging on audited Linux or UNIX computers

1. Switch to the root user.
2. Run the dadebug clear command to remove any existing detailed logging from previous operations.

   ```
   dadebug clear
   ```

3. Run the dadebug on command to enable detailed logging on for audit-related agent operations.

   ```
   dadebug on
   ```

   Detailed messages are recorded in the /var/log/centrifydc.log file. You can view the contents of the log file with a text editor. In most cases, however, you should collect additional information and send all of the logged information to Delinea Support.

4. Restart the auditing service.

   ```
   /usr/share/centrifydc/bin/centrifyda restart
   ```

5. Run the dainfo diagnostic command and save the output to a text file.

   ```
   dainfo --diag > /tmp/dainfo.txt
   ```

6. Run the adinfo diagnostic command and save the output to a text file.

   ```
   adinfo --diag > /tmp/adinfo.txt
   ```

7. Stop detailed logging of audit-related activity.

   ```
   dadebug off
   ```

8. Send an email to Delinea Support with the log files and the agent configuration file as an attachment.

   /var/log/centrifydc.log /tmp/dainfo.txt /tmp/adinfo.txt /etc/centrifyda/centrifyda.conf

To check whether detailed logging is enabled:

1. Run dadebug without parameters to see if detailed logging is currently enabled.

```
dadebug
```

Centrify DirectAudit debug logging is on.

2. Run addebug without parameters to see if detailed logging is currently enabled.

```
addebug
```

3. Run addebug off to disable logging, if needed.

Enabling Detailed Logging for the Collector Service

If you are troubleshooting an auditing-related issue, you should enable detailed logging for the collector service on the computers where the collector service runs.

To enable detailed logging on a collector:

1. Log on to a computer with a collector service.
2. Click **Start > All Programs > Centrify Server Suite 2021.1 > Centrify Audit & Monitoring Service > Audit Collector Control Panel** to open the Collector Control Panel.
3. Click the **Troubleshooting** tab.
4. Click **Options**, change the logging level to **Trace messages**, then click **Apply**.
5. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
6. Click **View Log** to view the current log file.

   From the log file window, you can also click File > Save As to save the log file.
7. Click **Close** to close the Collector Control Panel.
8. Send an email to Centrify Support with the log file from the location specified in Step 5 as an attachment.
9. Open the Collector Control Panel, click the **Troubleshooting** tab, click **Options**, change the logging level back to its default setting of **Informational messages**, then click **OK**.

Enabling Detailed Logging for Auditing Consoles

In most cases, troubleshooting auditing-related issues requires information about the operation of the agent and the collector or database activity. However, in some cases, it might be necessary to capture detailed information about the operation of Audit Manager or Audit Analyzer.

To capture detailed information for Audit Manager:

1. Log on to a computer with the Audit Manager console.
2. Click **Start > All Programs > Centrify Server Suite 2021.1 > Audit Manager** to open the Audit Manager console.
3. Select the Audit Manager node, right-click, then click **Log Settings**.
4. Change the logging level to **Trace messages**, then click **Apply**.
5. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
6. Send an email to Centrify Support with the log file from the location specified in Enabling detailed logging for the collector service as an attachment.
7. Right-click Audit Manager, click **Log Settings**, change the logging level back to its default setting of **Warning messages**, then click **OK**.

To capture detailed information for Audit Analyzer:

1. Log on to a computer with the Audit Analyzer console.
2. Click **Start** > **All Programs** > **Start > All Programs > Centrify Server Suite 2021.1 > Audit Analyzer**> to open the Audit Analyzer console.
3. Select the Audit Analyzer node, right-click, then click **Options**.
4. Change the logging level to **Trace messages**, then click **Apply**.
5. Note the log folder location or click **Browse** to specify a different location for the log file, then click **OK**.
6. Send an email to Centrify Support with the log file from the location specified in Enabling detailed logging for the collector service as an attachment.
7. Right-click Audit Analyzer, click **Options**, change the logging level back to its default setting of **Warning messages**, then click **OK**.

Tracing Database Operations

Database traces are used to help diagnose problems in the management database or audit store databases. For example, database traces can help to identify inconsistencies caused by hardware errors or network interruptions. After you enable database tracing, Audit Manager tracks all of the SQL statements and debug messages from the audit management database or audit store, and records the information in the database server.

> **Note**: Tracing database operations affects database performance. You should only activate a database trace if you require this information for troubleshooting. Before you start a database trace, try to reduce the load on the database instance as much as possible, then only perform the actions needed to reproduce the issue you are troubleshooting. Turn off database tracing as soon as you have logged the activity you need for the analysis of database operations. The trace for each database can take up to 800MB of server disk space. After you turn off database tracing, restart the SQL Server instance to reset the disk space.

Starting a Database Trace

You can start a database trace for a management database or an audit store database.

To start database tracing:

1. Open Audit Manager.
2. Select an installation name, right-click, then click **Properties**.
3. Click the **Database Trace** tab.

   This tab displays basic information about the management databases and audit store databases for the selected installation. In the Trace Status column, you can see whether tracing is enabled or disabled for each database.
4. Select a management or audit store database in the list, then click **Enable** to start tracing on the database selected.
5. Click **OK**, then perform the database actions for which you want to capture information.

Stopping the Database Trace

You should turn off database tracing immediately after you have logged the activity you need for the analysis of database operations.

To stop database tracing:

1. Open Audit Manager.
2. Select the installation name, right-click, then click **Properties**.
3. Click the **Database Trace** tab.
4. Select the management or audit store database that has tracing enabled, then click **Disable** to stop tracing on the database selected.
5. Click **Export** to save the database trace from the selected databases to a file with comma-separated values (.csv).
6. Follow the prompts displayed in the Export Database Trace wizard to save the information to a file.

Exporting the Database Trace for a Management Database

The Export Database Trace wizard prompts you for different information depending on whether the database trace is for a management database or an audit store database. For example, if you generate a database trace for a management database then click **Export**, the Export Database Trace wizard prompts you for user accounts.

To export the database trace:

1. Select a start date and time for the **From** filter and an end date and time for the **To** filter, then click **Next**.
2. Click **Add** to search for and select users, then click **Next**.

   By default, you can search for users in the entire directory, you can click **Object Types** or **Locations** to change the scope of the search scope, or click **Advanced** specify other criteria.
3. Accept the default folder location or click **Browse** to select a different location, then click **Next**.
4. Review your selections, then click **Next**.

By default, the wizard save the file as *installation_name*.csv and opens the file location.

5. Click **Finish**, then click **OK** to close the installation properties.

Exporting the database trace for audit store databases

When you select an audit store from the lower area of the **Database Trace** tab on the **Properties** page and click the lower **Export** button, the wizard opens with a date/time **Export Criteria** page. On the second page, the wizard asks you to pick the domain and computer.

To export the database trace:

1. Select a start date and time for the **From** filter and an end date and time for the **To** filter, then click **Next**.
2. Click **Add** to search for and select collectors, then click **Next**.

   By default, you can search for computers in the entire directory, you can click **Object Types** or **Locations** to change the scope of the search scope, or click **Advanced** specify other criteria.
3. Click **Add** to search for and select management database computers, then click **Next**.
4. Accept the default folder location or click **Browse** to select a different location, then click **Next**.
5. Review your selections, then click **Next**.

   By default, the wizard save the file as *audit_store_name*.csv and opens the file location.
6. Click **Finish**, then click **OK** to close the installation properties.

Delegating Database Trace Management

You can delegate the authority to manage database tracing by granting the Manage Database Trace permission to other users for a management database or an audit store database.

Stopping Auditing on a Computer

Several actions can directly or indirectly stop auditing on a computer. For example:

- Someone powers down the audited computer.
- Someone logs in on the audited computer and stops the agent.
- The audited computer is moved to a different audit store, causing the initial audit store to consider the audited computer disconnected.
- The administrator checks the **Define trusted audited computer** list on the Advanced tab of an Audit Store Properties page, and does not include the audited computer on that list.

Resuming Auditing if the Agent Stops

If the dad service stops running for any reason, audited shell sessions will stop working and you will be prompted to resume or quit auditing. If you resume auditing, the cdawatch process attempts to start dad and connect to the installation. However, if you have manually stopped the dad process, for example by running /usr/share/centrifydc/bin/centrifyda stop, you must manually restart the agent.

If you decide to quit auditing when the dad service has stopped running, you are prompted to confirm that you want to terminate the session before the session ends.

Allowing Users to Log in when Auditing is Stopped

If auditing is required but the agent is not running, users might be prevented from logging in. You can log in as a user with root privileges and either restart the agent or temporarily disable auditing using `dacontrol -d` to allow users to log in.

You can also run `dainfo --diag` or check the log file to get more information. For example, if the adclient process is not running, you might be unable to restart auditing.

If you cannot immediately correct the problem, you can temporarily disable all auditing.

Determining Collector Status and Connectivity

You can use the Collector Control Panel to generate a complete diagnostic check of the collector. The diagnostic report includes detailed information about the current status of the collector and the installation and audit store to which the collector sends data.

To generate diagnostics on a collector:

1. Log on to a computer with a collector service.
2. Click **Start > All Programs > Centrify Server Suite 2021.1 > Centrify Audit & Monitoring Service > Audit Collector Control Panel** to open the Collector Control Panel.
3. Click the **Troubleshooting** tab.
4. Click **Diagnostics**.

The results display in a Diagnostic Information window. If connections are successful and components are configured correctly, you should see results similar to this:

```
Establishing connection with Collector: Success
 Getting collector's current status: Running
 Getting Collector's current Installation: DefaultInstallation (locally configured)
 Getting Collector's current Audit Store: Data Source=pysql.py.dev\\CENTRIFYSUITE;Initial Catalog=AuditStoreWindows-2018-11-02
 Machine IP address(es): 10.140.16.59
 Machine is joined to: py.dev
 Forest: py.dev
 Using Domain Controller: pydc.py.dev
 Is Global Catalog Available: True
 Using Global Catalog: pydc.py.dev
 Machine is in site: Default-First-Site-Name@py.dev
 Installations:
 DefaultInstallation
 AD Object: py.dev/Program Data/Centrify/DirectAudit/Vegas-Installation-d97d8fc9-7876-4f5b-b161-4a7b3736b8ec
 Object GUID: 1563b2e1-1ea3-4307-ac51-7c92fdf5cb8a
 Installation ID: f7b36b73-0384-4283-b6f7-63a1cdb77b17
 Audit Stores:
 AuditStoreUNIX
 Site(s): (Default-First-Site-Name@py.dev)
 Subnet(s): None configured
 Affinity: UNIX
 Trusted Agents: None configured
 Trusted Collectors: None configured
 Audit Store Active Database:
 Data Source=pysql.py.dev\\CENTRIFYSUITE
 Initial Catalog=AuditStoreUNIX-2018-11-02
 Additional Connection Parameters=\<none\>
 AuditStoreWindows
 Site(s): (Default-First-Site-Name@py.dev)
 Subnet(s): None configured
 Affinity: Windows
 Trusted Agents: None configured
 Trusted Collectors: None configured
 Audit Store Active Database:
 Data Source=pysql.py.dev\\CENTRIFYSUITE
 Initial Catalog=AuditStoreWindows-2018-11-02
 Additional Connection Parameters=\<none\>
 Machine's Installation: DefaultInstallation (locally configured)
 This machine's Audit Store is 'AuditStoreWindows' based on preferred Audit Store (locally configured)
 Attempting to connect to Audit Store:
 Data Source=pysql.py.dev\\CENTRIFYSUITE
 Initial Catalog=AuditStoreWindows-2018-11-02
 Integrated Security=TRUE
 Pooling=True
 Max Pool Size=1000
 Encrypt=True
 TrustServerCertificate=True
 Additional Connection Parameters=\<none\>
 Connected to Audit Store successfully

 Done.
```

You can copy the results to a file and send them to Centrify Support for help.

Resolving Connectivity Issues between a Collector and an Audit Store

If the diagnostic report or the Collector Configuration wizard indicates that the collector cannot connect to an audit store database, check the following:

- Verify the account you logged in with has permission to add a collector.

- Verify the collector service has permission to connect to the active audit store database. You can grant this permission from Audit Manager.
- Check whether the SQL Server instance needs to be restarted. For example, make sure the SQL Server instance is not waiting for a restart to complete ASP.NET registration changes.
- Check whether there is a firewall between the collector and the SQL Server instance blocking access.
- Check whether SQL Server is configured to allow named pipes and TCP/IP connections.
- Check whether SQL Server is configured to allow remote connections.
- Compare the site or subnet that the collector is configured to use with the scope of the audit store. For example, make sure the audit store site or subnet matches the site or subnet in the audit store properties.

  AuditStore
  Site(s): (Default-First-Site-Name@pistolas.org)
  Subnet(s): None configured

Resolving Authentication Issues

If you configure the collector service to use an Active Directory account instead of the local system account, you might encounter problems with Kerberos authentication when the collector attempts to connect to the audit store database. Kerberos authentication uses the service principal names (SPN) registered for the SQL Server account to authenticate a service. When the collector (client) wants to connect to SQL Server, it locates an instance of the service, composes an SPN for that instance, connects to the service, and presents the SPN for the service to authenticate. If the collector service account does not have any SPNs, the Kerberos authentication request fails.

To resolve this problem, go to KB-1311 in the Centrify Knowledge Base, select **Attachments**, and click **View > Open > Run** to run the checkspn.vbs script on a computer that is joined to Active Directory.

> **Note**: The user who is running this command must have permission to register the SPN on the service account.

By default, this script runs in report-only mode. It checks whether the required SPNs are present on the service account in question and issues a prompt to fix it, if not. This script registers the SPN in the service account servicePrincipalName attribute in the format:

MSSQLSvc/<FQDN>:<tcpport>

Monitoring Collector Performance Counters

If you have enabled auditing and installed the collector service on a local Windows computer, you can add audit-specific performance counters to Performance Monitor to help you analyze and resolve audit-related issues. When you install the collector, the performance counters are added automatically, if you uninstall the collector, the counters are also automatically removed from Performance Monitor.

To add Server Suite performance counters:

1. Log on to a computer with a collector service.
2. Click **Start > Administrative Tools > Performance Monitor**.
3. Expand Monitoring Tools and select **Performance Monitor**.
4. Click the green plus (+) icon in the toolbar.
5. Find the Audit Collector from the list, and expand it to show the list of available performance counters.

   The performance counters generally fall into one of three categories; agent information, packet volume, and data loads. For example, if you add the counter # Connected Agent, you will be able to view the number of agents currently connected. If you add the counter # Unix Meta Message Packet, you will be able to view the number of Unix meta message packets. If you add the counter, Bytes Unix Command, you will be able to view Unix command data in bytes.
6. Choose the performance counter you would like to add and click **Add**.
7. Repeat Step 6 until you have added the counters you want to monitor.
8. Click **OK**.

Managing Microsoft SQL Server Databases

Managing an audit installation requires permission to create new SQL Server databases on a SQL Server instance. In a production environment, this is an ongoing process to keep databases small and efficient. Because the management of the audit databases is not a onetime setup operation, Delinea recommends that you have at least one dedicated SQL Server instance for the audit administrator to use. The audit administrator should also be a member of the SQL Server system administrator role to ensure full control over the databases created and archived.

Selecting SQL Server or Windows Authentication

When you configure the Microsoft SQL Server instance to use for auditing, you must specify the type of authentication to use. The appropriate type of authentication depends on how your production environment is configured. For example, if you have a firewall between components or one-way trust relationship between forests, you must allow SQL Server authentications.

To support the auditing infrastructure, you can use the following types of authentication:

- Windows authentication for creating new databases.
- Windows authentication or both SQL Server authentication and Windows authentication for connections between collectors and audit stores.
- Windows authentication or both SQL Server authentication and Windows authentication for connections between audit stores and the audit management database.
- SQL Server authentication for collectors in an untrusted forest and an audit store in a trusted forest.
- SQL Server authentication for audit store databases in a trusted forest and audit management database in an untrusted forest.

If you choose Windows authentication, you can perform actions with your own logon account or using another Windows account name and password.

Connecting to an Installation or Database

If you unable to connect to the SQL Server database, the problem might be caused by one of the following issues:

- A firewall blocking access to the SQL Server instance.
- TCP/IP has not been enabled for the SQL Server instance of SQL Server
- Remote connections have not been enabled for the SQL Server instance.

For information about areas to check, see the following article:

https://blog.sqlauthority.com/2009/05/21/sql-server-fix-error-provider-named-pipes-provider-error-40-could-not-open-a-connection-to-sql-server-microsoft-sql-server-error/

Assigning the Service Principal name for SQL Server

If you get error messages when performing database operations, such as creating a new audit management database using Audit Manager, the problem is likely because the service principal name (SPN) for the SQL Server instance is assigned to the wrong Active Directory container.

- If the SQL Server startup account is a local system account, the appropriate container is the computer name.
- If it is any other account, the appropriate container is the SQL Server startup account.

Because authentication tries to use the first SPN it finds, make sure that no SPNs are assigned to inappropriate containers. Usually this error occurs when the administrator does not remove a manually added SPN from the Active Directory container after changing the SQL Server service account.

For help troubleshooting this problem, read the following article:

https://support.microsoft.com/en-us/help/811889/how-to-troubleshoot-the-cannot-generate-sspi-context-error-message

Publishing Installation Information in Active Directory

The default location for publishing audit installation information in Active Directory is:

domain/Program Data/Centrify/DirectAudit

In most cases, this location is accessible to any administrative user. If you cannot access the publication location, check the following:

- Make sure you have permission to publish information to Active Directory.

- Verify that the publication location exists in Active Directory.
- Check the network for problems.

Moving a service connection point from its published location can result in connection problems. If you delete the default publication location and add a new publication location, you might not have permissions on the new location. If you do not have the appropriate permissions on the new location, ask the Active Directory administrator to grant you such permissions before running any of the wizards to reconfigure agents and collectors.

> **Note**: A new location might not be reflected immediately in the list current published locations. However, this has no any adverse effects apart from not being able to see the published location.

Monitoring File System Disk Space Usage

Like most software applications, Centrify Agents require adequate disk space to be available to operate properly. For example, agents read and write temporary files to authenticate processes and ensure data integrity. If your operating system does not have enough disk space to accommodate these temporary files, the agent might be unable to run and prevent users from logging on or activity from being audited.

To prevent problems with disk space allocation, you should monitor key directories, such as the /tmp and /var directories, to ensure free space is available. The disk space required by different directories depends on the configuration and operating systems of the computer and the Active Directory environment. However, if any directory approaches 100% of its allocation, you should allocate more disk or remove older files to free up space for continued operation.

Command Line Programs for Managing Audited Sessions

This chapter provides an overview of the command line interface that you can use to manage audited computers. For complete reference information about the required and optional parameters for each command, see the man page provided locally on the Centrify-managed computer.

How to Use Command Line Programs

Command-line programs allow you to perform administrative tasks directly from a UNIX shell or by using a shell script. These programs are installed when you install the Centrify Agent for *NIX, and are installed by default in the following directories:

- /usr/sbin
- /usr/bin

You can use the UNIX command-line programs to take action directly on a local UNIX computer, for example to enable or disable auditing manually on a local computer. You can also use these programs to perform administrative or diagnostic tasks when it is more convenient to run them on the UNIX computer than through Audit Manager. For example, you might find it more convenient to view details about the agent configuration or diagnostic information directly on a local computer rather than through Audit Manager or the Agent Control Panel.

Displaying Usage Information and Man Pages

You can display a summary of usage information for any UNIX command-line program by typing the command plus the --help or -h option. For example, to see the usage information for the dacontrol command:

```
dacontrol --help
```

For more complete information about any command, read the command's man page. For example, to see the man page for the dacontrol command, type:

```
man dacontrol
```

Using Commands for Administrative Tasks

The command-line programs allow you to perform administrative tasks—such as enable or disable shell auditing on UNIX computers or generate diagnostic information—directly on an audited computer. The following table provides a summary of the auditing-related programs installed with the Centrify Agent for *NIX and the Centrify Client for Linux audit package. For complete information about the syntax and options for any command, see the man page for that command.

dacheck	The dacheck command performs operating system, network, and Active Directory tests to verify a computer meets the system requirements for a successful installation. For example, the install.sh script runs the dacheck program. The dacheck command is located in the same place as the adcheck command: /usr/share/centrify/dc/bin.
dacontrol	Enable or disable session or individual command auditing on a computer. You can also use this command to manually configure the audit installation to use for a local computer if you are not identifying the installation by group policy. Only users with root privileges can run the dacontrol command. Note: {/b}If the audited system is not joined to Active Directory and it is audited by way of the Centrify Client for Linux, you cannot change the audit installation with the dacontrol command.
dad	Start the dad process manually. The dad process records terminal activity on the UNIX computer and transfers the data to a collector. In most cases, it is automatically started when the computer is first booted. However, you can run this command to manually start the audit process on a local computer. Only users with root privileges can run the dad command.
dadebug	Enable or disable logging for the dad process on an audited computer. If you enable logging, the dad process writes messages to the /var/log/centrifydc.log file. If you run dadebug without specifying an option, the command returns a status message that indicates whether logging is currently enabled or disabled. Only users with root privileges can run the dadebug command.
dadiag	Display detailed information about the configuration and current auditing status for a local computer. This command displays the same information as dainfo --diag.
	Clear the auditing service in-memory cache of name service queries and installation information. If you run this command without any

daflush	arguments, it removes both auditing-related name service query results and audit installation information from the in-memory cache. If you run this command with no arguments or specify the nameservice option, the command also automatically clears the cache for common name services—such as nscd and pwgrd—if those services are running on the local computer. Clearing the cache of name service query results is useful if you make changes that would affect the results of a name service query, and want to ensure you get updated information. For example, if you remove the UNIX Login role for an Active Directory user, some information for that user might remain in the auditing service cache and be returned when you run a command such as getent passwd for that user. You can run daflush to ensure the user is removed completely from the local computer cache, including the auditing service cache. Only users with root privileges can run the daflush command.
dainfo	Display detailed information about the status and configuration of an audited computer.
dareload	Force the dad process to reload configuration properties from the /etc/centrifyda/centrifyda.conf file or the advanced monitoring properties from /etc/centrifyda/libaudit.conf. This command enables you to apply configuration changes without restarting the agent. Only users with root privileges can run the dareload command.
dashellfix.sh	Reset shells to their source shell on computers that are not being audited in an audited zone. On audited computers, the cdash shell is used to capture and forward audit data instead of the original shell. This script enables you to restore the user's original shell choice if the auditing service and wrapper shell are removed.
daspool	Display information about the size and content of the auditing-related offline cache (spool) files. If an audited computer cannot contact a collector service, it caches session, audit trail, and other information locally until a collector becomes available. This command enables you to review information about these offline cache files. Only users with root privileges can run the daspool command.

Configuring Duplicate Audit Session Cleanup

Sometimes the auditing service records duplicate sessions if your auditing installation includes one or more UNIX computers where both of the following situations occur:

- The DirectAudit agent is installed.
- A user can log in to the computer from the Admin Portal and the cloud tenant is enabled for auditing.

To avoid this situation, add the following environment variable to your /etc/centrifydc/ssh/sshd_config file:

```
AcceptEnv centrify_cip_da_data
```

Note that the above /etc/centrifydc/ssh/ path applies if you're using the Centrify OpenSSH server. If you're using a different SSH server, the file path may be different-- so be sure to update the appropriate SSH daemon configuration file for your system.

With the environment variable set, the agent uses that to verify the SSH public key of the associated tenant. That way the auditing service can determine which sessions are duplicated and remove them. Also, the agent on the UNIX computer will no longer record sessions that originate from the Admin Portal on the same computer.

Downloading the Tenant SSH Public Key

There's a script called dadownloadsshpublickey.tcl that downloads the tenant's SSH public key. With the public key and the centrify_cip_da_data environment variable, the auditing service can determine which audit sessions are duplicates and remove them.

The agent installer puts this tcl script into /usr/bin, except for CoreOS systems where the installer puts the script into /opt/centrify/bin. This script requires root privilege to run. The output file specified by dad for the script is /var/centrifyda/tenant_rsa.pub.

If da fails to download the public key or if you need to change the public key after da has started, you can manually run this tcl script.

/usr/bin/dadownloadsshpublickey.tcl --output-file /var/centrifyda/tenant_rsa.pub

Use the following options when you run this script:

- --cip, --i *‹cloud tenant URL›*

This option is optional.

If the computer is not joined to the domain currently, use this option to specify the cloud tenant URL. If you don't use this option, the script finds the URL automatically if the computer is joined to the domain.

- --output-file, -o *<file>*

  This option is required.

  Use this option to specify the output filename for the tenant's SSH public key. This file must be in a parent directory that is writable by root only and the directory cannot be a symlink.

Installing the UNIX Agent on Remote Computers

In most cases, you install the UNIX agent locally on a computer using the install.sh script interactively. You can install the UNIX agent on remote computers using the install.sh script and a configuration file or using virtually any software distribution or package installer program. This chapter provides an overview of these alternatives for installing the agent on UNIX or Linux computers.

Installing the Agent Silently using a Configuration File

You can automate agent installation by running the install.sh script in non-interactive mode:

```
install.sh -n
```

In this mode, the script uses configuration details specified in the centrifyda-install.cfg file. If this file is not found, the install.sh script uses its built-in default values.

To specify configuration values, edit the sample centrifyda-install.cfg file in its default location, or create a new text file with the same name, and then run the install.sh script.

In the file, INSTALL=Y installs the agent, and INSTALL=U upgrades the agent.

By default, the script returns an exit code of 0 if the operation is successful. To return exit codes that provide more detailed information about the result, use:

install.sh -n --custom_rc

CODE_SIN=0	Successful install
CODE_SUP=0	Successful upgrade
CODE_SUN=0	Successful uninstall
CODE_NIN=24	Did nothing during install
CODE_NUN=25	Did nothing during uninstall
CODE_EIN=26	Error during install
CODE_EUP=27	Error during upgrade
CODE_EUN=28	Error during uninstall
CODE_ESU=29	Error during setup; for example, unsupported operating environment or invalid arguments

Using Other Programs to Install the UNIX Agent

Auditing-related files are bundled with the core Centrify Agent files into a platform-specific software package. You must install the Centrify Agent on the audited computer before you enable the auditing service.

To install auditing using a native installation mechanism:

1. Log on as a user with root privileges.
2. If you want to install from a CD and the drive is not mounted automatically, use the OS-specific command to mount the cdrom device.
3. Copy the appropriate package to a local directory.

   For Solaris 10:

   cp /cdrom/cdrom0/Unix/centrifyda-*n.n.n*-sol10-sparc-local.tgz .

For Red Hat Enterprise Linux:

cp /mnt/cdrom/Unix/centrifyda-*n.n.n*-rhel5-x86_64.rpm .

For SuSE Linux:

cp /mnt/cdrom/Unix/centrifyda-*n.n.n*-suse11-x86_64.rpm .

4. If the software package is a compressed file, unzip and extract the contents. For example, on Solaris:

   gunzip -d centrifyda-*n.n.n*-sol10-local.tgz
   tar -xf centrifyda-*n.n.n*-sol10-sparc-local.tar

5. Run the installation command appropriate to the operating environment.

   For Red Hat Linux, you can use:

   rpm –ivh centrifyda-*n.n.n*-rhel5-x86_64.rpm

   For SuSE Linux, you can use:

   rpm –ivh centrifyda-*n.n.n*-suse11-x86_64.rpm

   For Solaris, you can use:

   pkgadd –d CentrifyDA -a admin

   Note: You can also use other programs, such as SMIT or YAST, to install the agent package.

6. If you are using an installation with a name other than DefaultInstallation, you need to configure it with dacontrol or using group policy.

   If there is an installation with the name DefaultInstallation the UNIX agent uses it by default. For more information about specifying the installation, see Configuring the installation for an agent.

7. After installing the package, use dainfo to verify that auditing is installed and running. You should see output similar to the following:

```
Pinging adclient: adclient is available
    Daemon status: Online
    Current collector: DC2008r2-LG.pistolas.org:
    5063:HOST/dc2008r2-lg@PISTOLAS.ORG
     Session offline store size: 0.00 Bytes
     Session despool rate: 0.00 Bytes/second
     Audit trail offline store size: 0.00 Bytes
     Audit trail despool rate: 0.00 Bytes/second
     Getting offline database information:
     Size on disk: 52.00 KB
     Database filesystem use: 3.06 GB used,
     15.52 GB total, 12.45 GB free
     DirectAudit NSS module: Active
     User (root) audited status: Yes
     DirectAudit is not configured for per-command auditing.
```

Permissions Required to Perform Administrative and Auditing Tasks

This section describes the permissions required to perform various auditing-related activities.

Setting and Synchronizing Audit-related Permissions

As a Master Auditor, you can set the permissions that control what all other administrators and auditors can do. In most cases, you set these permissions by making selections in Audit Manager. Your selections are saved in the management database for each installation, then published in Active Directory whenever you synchronize the management database with the service connection point for the installation.

The permissions you can set consist of a specific action that can be taken, a scope to which the action applies, and the specific Active Directory user or group to which you are granting the permission.

For example, a permission might specify an action, such as ability to modify a name or detach a database with a scope such as a specific installation or audit store database. For each action and scope, you select the Active Directory user or group to be granted that permission. After users or groups are granted a permission, they are called a trustee for that action and scope.

To view the existing permissions, right-click an installation or an audit store and select Properties, then click the Security tab.

Component by Component Permissions

The table below lists the permissions needed to create or add to an installation one component at a time.

Create an audit installation	
Create an audit console	
Create a SQL Server instance	
Check a SQL Server service account	
Add a service connection point	
Add a publication location	Audit server administrator or Manage Publication Locations (Installation)
Add a UNIX agent to an audited machine	
Add a Windows agent to an audited machine	
Enable trusted audited machine list for an audit store	Audit server administrator or Manage Collectors (Installation)
Add an audited machine to the trusted list for an audit store	Audit server administrator or Manage Collectors (Installation)
Add a collector	[does not require any special permissions to install]
Enable trusted collector list for an audit store	Audit server administrator or Manage Collectors (Installation)
Add a collector to the trusted list for an audit store	Audit server administrator or Manage Collectors (Installation)
Add an audit store	Audit server administrator or Manage Audit Store List (Installation)
Add an audit store database	SQL: Database owner (dbo) or a delegated member of the db_owner role or Audit store administrator (Installation) or Audit server administrator (Installation) or Manage Databases (Installation)

Attach an audit store database Change which DB is active Attach DA version 1 database	Audit Store administrator (Installation) or Audit server administrator (Installation) or Manage Databases (Installation)
Change which DB is active	Audit Store administrator or Audit server administrator or Manage Databases
Add a subnet or AD site to the audit store	Audit Store administrator or Audit server administrator or Manage Sites (Audit store)
Add an audit server	Manage Audit Server List (Installation)
Add an audit role; change its definition, menbership or permissions	Creator of installation (Installation) or Audit server administrator (Installation) or Manage Audit Roles (Installation)

Installation Permissions

Installation permissions allow users or groups to modify different aspects of an installation's properties. By default, the Master Auditor and the management database administrator have Full Control over the installation and can assign the following permissions to other users and groups:

Full Control	Perform all administrative tasks on the selected installation and assign permissions to other users and groups.
Change Permissions	Add, modify, or remove Active Directory users and groups that have specific permissions. A user or group granted this permission can display the properties for the installation, then click the **Security** tab to select permissions for other users and groups.
Modify Name	Modify the name of the selected installation. A user or group granted this permission can display the properties for the installation, then click the **General** tab to change the installation name.
Manage Management Database List	Add or remove a management database for the selected installation. A user or group granted this permission can right-click the installation name in Audit Manager and select **Management Databases** to add or remove a management database. Deleting the management database from Microsoft SQL Server requires additional SQL Server permissions.
Manage Audit Store List	Add, modify, or remove audit stores and audit store databases for the selected installation. A user or group granted this permission can use the Add Audit Store wizard or right-click the installation name in Audit Manager, select **Management Databases**, then click **Properties** to add or remove sites or subnets associated with the installation.
Manage Collectors	Add, modify, or remove collectors for the selected installation.
Manage Audited Systems	Add, modify, or remove audited computers for the selected installation.
Manage Audit Roles	Add, modify, or remove audit roles for the selected installation.
Manage Queries	Add, modify, or remove queries for the selected installation.
Manage Publications	Add, modify, or remove publication locations in Active Directory for the service connection point associated with the selected installation. A user or group granted this permission can display the properties for the installation, then click the **Publication** tab to change the publication location in Active Directory for the installation. A user or group granted this permission can also update the information stored in Active Directory to keep the information in Active Directory synchronized with the information stored in the management database. However, users or groups with this permission must have sufficient Windows rights to be able to update objects in Active Directory.

Manage License	Add or remove license keys for an installation. A user or group granted this permission can display the properties for the installation, click the **General** tab, then click **Details** to manage licenses for the installation.
Modify Notification	Enable or disable the audit notification message for the selected installation. A user or group granted this permission can display the properties for the installation, then click the **Notification** tab to manage the notification message and image for the installation.
Modify Audit Options	Enable or disable video capture auditing for the selected installation. Control whether users are allowed to update the review status of their own sessions. Control whether users are allowed to delete their own sessions. A user or group granted this permission can display the properties for the installation, then click the **Audit Options** tab to manage installation-wide auditing options.
View	Enable read-only permission for the selected installation. If a user has only View permission, they can see all the auditing components in the Audit Manager console, but they do not have access to audited sessions nor can they change any installation details.

Setting Installation Permissions

You can set installation permissions for a specific installation, by selecting the installation name in Audit Manager.

To set permissions on an installation:

1. Open Audit Manager and select the installation name.
2. Right-click, then click **Properties**.
3. Click the **Security** tab.
4. Click **Add** to open Select Users and Groups.
5. Type the user or group name who should be granted installation permissions, then click **OK**.

   You can add multiple users or groups from the Select Users or Groups dialog box. You can also type part of the name, then click **Check Names** to look up user and group names.
6. Select the specific permissions you want to grant to the selected user or group.

Management Database Permissions

Management database permissions allow users or groups to modify different aspects of an installation's management database. By default, the Master Auditor and the management database administrator have Full Control over the management database and can assign the following permissions to other users and groups:

Full Control	Perform all administrative tasks on the selected management database and assign permissions to other users and groups.
Change Permissions	Add, modify, or remove Active Directory users and groups that have specific permissions. A user or group granted this permission can display the properties for the management database, then click the **Security** tab to select permissions for other users and groups.
Modify Name	Modify the name displayed for the selected management database. A user or group granted this permission can display the properties for the management database, then click the **General** tab to change the management database name.
Manage Scopes	Add, modify, or remove sites or subnets for a management database. A user or group granted this permission can display the properties for the management database, then click the **Scope** tab to add or remove sites and subnets.
Remove Database	Remove a management database from an installation. Deleting the management database from Microsoft SQL Server requires additional SQL Server permissions.

Manage SQL Logins	Add or remove the Allowed incoming users for the selected management database. A user or group granted this permission can display properties for the management database, then click the **Advanced** tab to add or remove allowed accounts, or to change the outgoing account or authentication type.
Manage Database Trace	Enable, disable, or export database traces for the selected management database.

Setting Management Database Permissions

You can set management database permissions for a specific installation, by selecting the installation name in Audit Manager.

To set permissions on an management database:

1. Open Audit Manager and select the installation name.
2. Right-click, then click **Management Databases**.
3. Select the management database, click **Properties**, then click the **Security** tab.
4. Click **Add**, type the user or group name who should be granted permissions, then click **OK**.
5. Select the specific permissions you want to grant to the selected user or group.

Audit Store and Audit Store Database Permissions

Audit store permissions allow users or groups to modify different aspects of an audit store or audit store database. By default, the Master Auditor and the audit store database administrator have Full Control over the audit store and its database and can assign the following permissions to other users and groups:

Full Control	Perform all administrative tasks on the selected audit store database and assign permissions to other users and groups.
Change Permissions	Add, modify, or remove Active Directory users and groups that have specific permissions. A user or group granted this permission can display the properties for the audit store, then click the **Security** tab to select permissions for other users and groups.
Modify Name	Modify the name displayed for the selected audit store. A user or group granted this permission can display the properties for the audit store, then click the **General** tab to change the audit store name.
Manage Scopes	Add, modify, or remove sites or subnets for the audit store. A user or group granted this permission can display the properties for the audit store, then click the **Scope** tab to add or remove sites and subnets.
Manage SQL Logins	Add or remove the allowed incoming collectors and management database logins for the selected audit store database. A user or group granted this permission can display properties for the audit store database, then click the **Advanced** tab to add or remove accounts for collectors and management databases.
Manage Collectors	Add, modify, or remove trusted collectors for the audit store. A user or group granted this permission can display properties for the audit store, then click the **Advanced** tab to add or remove accounts trusted collectors.
Manage Audited Systems	Add, modify, or remove trusted audited computers for the audit store. A user or group granted this permission can display properties for the audit store, then click the **Advanced** tab to add or remove accounts trusted audited computers.
Manage Databases	Add, attach, detach, or delete audit store databases for the selected audit store.
Manage Database	Enable, disable, or export database traces for the selected audit store.

Trace

Audit Role Permissions

Audit role permissions allow users or groups to modify different aspects of an audit role. By default, the Master Auditor has Full Control over the audit roles and can assign the following permissions to other users and groups:

Full Control	Perform all administrative tasks on the selected audit role and assign permissions to other users and groups.
Change Permissions	Add, modify, or remove Active Directory users and groups that have specific permissions. A user or group granted this permission can display the properties for the audit role, then click the **Security** tab to select permissions for other users and groups.
Change Role Membership	Add, modify, or remove Active Directory users and groups that are assigned to the selected role. A user or group granted this permission can use the Add Audit Role wizard to assign users and groups to an audit role or select an audit role name, right-click, then select **Assign Users and Groups** to modify the role membership.
Change Role Definition	Modify the name, description, access, or privileges for the selected audit role. A user or group granted this permission can display the properties for the audit role, then: Click the **General** tab to modify the role name or description. Click the **Access** tab to modify the type of session and other criteria. Click the **Privileges** tab to modify hat users and groups assigned to the role can do.

Auditor Permissions

Auditor permissions allow users or groups to view, create, share, and delete queries. For an installation, the Master Auditor can control access to Audit Analyzer and queries using the Manage Queries permission and the assignment of audit roles. The privileges associated with an audit role also control whether auditor can update the review status or replay sessions. By default, the Master Auditor has Full Control over the auditor permissions and audit roles and can assign the following permissions to other users and groups:

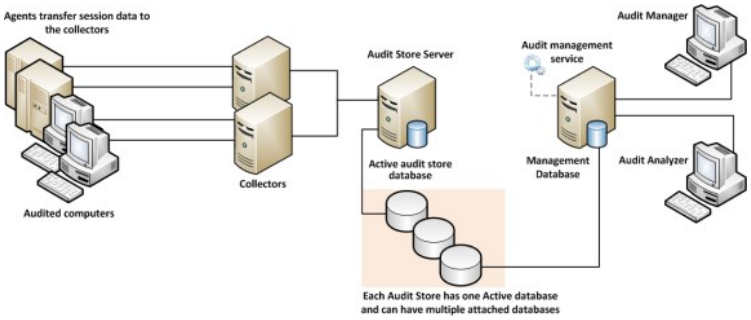
Full Control	Perform all administrative tasks on the selected query and assign permissions to other users and groups.
Change Permissions	Add, modify, or remove Active Directory users and groups that have specific permissions. A user or group granted this permission can display the properties for the query, then click the **Security** tab to select permissions for other users and groups.
Read	Read the selected query definition, session results, and indexed commands.
Delete	Delete the selected query definition, session results, and indexed commands.
Modify	Modify the selected query definition, session results, and indexed commands.

Sizing Recommendations for Audit Installations

A typical deployment of Centrify Audit & Monitoring Serviceconsists of a number of components such as one or more audited Systems (UNIX/Linux or Windows), one or more collectors, audit management server, management database, one or more audit store databases and consoles (the Audit Manager and the Audit Analyzer consoles) which all communicate with each other. Given the complexity of this communication and number of components involved, good planning is important for a successful deployment of the product. When planning a deployment, some of the most common questions that we asked are below:

- Will just one installation of Centrify Audit & Monitoring Service suffice? Or are multiple installations needed or recommended?
- How many audit stores need to be provisioned in each of the installations and how should their scope be configured?
- How many collectors will be needed and what kind of hardware is recommended for each of them?
- What is the recommended version/edition of SQL Server and what kind of hardware is recommended to host this SQL Server?

You must take into consideration a number of factors when deciding how to plan and configure the audit and monitoring servicedeployment and what kind of hardware will be needed to deploy the key components. This section will help you understand these factors in detail and come up with answers to such questions.



Planning an Audit and Monitoring Service Deployment

System Integrators often rely on the number of audited systems to estimate the hardware requirements and to come up with the overall strategy of audit and monitoring service deployment. For example, an environment with 100 audited systems may look like a small setup and one may incorrectly conclude that it's a small scale deployment that won't require a powerful hardware to support it. Once setup however, such assumptions may turn it into a deployment that seldom scales and often produces poor performance, both when capturing the audit activity and when querying the already captured audit data.

Below are a few factors that you must consider before making any deployment decisions,

SQL Server

Out of all the components in the audit and monitoring service ecosystem, SQL is the most heavyweight and will share most of the burden when it comes to workload. Using a properly equipped and optimally configured SQL Server is very important. The version and edition of SQL Server being used (such as Express or Standard or Enterprise) or the type of machine being used to host the SQL Server (such as a virtual or physical machine) can noticeably improve the overall performance. On the contrary, a poorly configured SQL Server may produce a very poor performance no matter how powerful the underlying hardware is.

Number of Concurrently Audited Users

Relying on the number of audited systems is not always a good assumption. For example, an environment may have just a handful of systems but may have a large number of users logging into these systems on a daily basis. A jumpbox scenario such as Citrix XenApp Server is a perfect example. When planning, you should plan for the number of concurrently audited users, not just the total number of audited systems. User activity patterns and behaviors also play an important role in overall performance and storage requirements. For example, the audited data will be much smaller in an environment where no logins are expected most of the time as compared to a network control systems wherein audited users are logging on and logging out throughout the day. The sizing guidelines specified in the later section of this whitepaper have all been based on workload simulations for the exact same reason.

What Needs to be Captured

What's being captured controls the overall workload on various components. Capturing video is more expensive than not doing so in terms of disk usage and load on collectors and SQL Server. Similarly, capturing interactive sessions is always going to produce more audited data when compared to capturing a

handful of commands thus putting system under more pressure. Capturing large quantities of data has another side effect; it slows down database backups and other maintenance processes which is not always liked by the database administrators.

Who Needs to be Audited

Who is being audited is equally important. Under default settings, the audit and monitoring service audits everything and everybody and this may not be a practical solution in many large environments. In production environments, it's very common to see processes or scheduled tasks that periodically monitor UNIX/Linux or Windows systems for their health by remotely executing certain commands (System Monitoring and Management software, such as BMC Patrol that periodically runs vmstat or iostat command on each of the UNIX/Linux systems is a good example). Activities like these needlessly generate thousands of Audited sessions on a daily basis and in many cases create tremendous load on an entire audit and monitoring service system.

UNIX/Linux and Windows

The type of system being audited influences the amount of data that will be captured from that system and the overall CPU load on collectors. For example, a Windows audited system almost always generates more data per day compared to a UNIX audited system with comparable number of concurrent users. This also means that an environment with Windows audited systems will most likely be more demanding (in terms of hardware resources) compared to an environment with same number of UNIX/Linux audited systems.

Query Performance

Query performance is one factor that often gets ignored. Capturing user activity and storing it in the database in a reasonable time is important. What's also important is to be able to search these records in a predictable time frame irrespective of the combined size and number of all the databases in the Centrify Audit & Monitoring Service system.

Audit Data Retention Policy

Audit data retention policy dictates how many days of data should be online and readily available for querying purpose and this number varies from one enterprise to another. Pay special attention to data retention policy requirements in the target environment. A longer retention policy typically results in large databases which also suffer from poor query performance if databases are not well maintained. On the contrary, too frequent rotation will also result in poor query performance if you keep too many inactive databases attached to the audit store.

System Overheads

Keep in mind the overhead that is caused by the Centrify Audit & Monitoring Service system itself; there are a number of background jobs carried out by various components of the audit and monitoring service system, including the audited systems themselves, collectors, and the Audit Management Server. This includes activities such as sending the audited system's heartbeat to the database (by way of collector), sending the collector's heartbeat to the database, processing active sessions list, processing and synchronizing information of audit roles with Active Directory Group criteria, calculating effective size of audited sessions, storing license usage information in Active Directory, and many more.

Latency

Geography/Network topology play an important role as it introduces latency. For example, an environment may well have just a handful of audited systems but if they're not geographically co-located, you may see delays in getting the audited user activity to its final destination (the database server); the same may happen if audited systems are not connected to collectors by a network link with reasonable bandwidth. A general rule of thumb is to group together audited systems, collectors and databases that are connected by a high speed network using the concept of audit store.

Best Practices for an Audit Installation

The previous section listed out a number of factors that may affect how a audit and monitoring service system will be deployed. Below is a set of best practices that are derived from these factors. Follow these practices for planning any audit and monitoring service deployment (large or small). You can also refer to the last section of this whitepaper that discusses how to tweak settings in an existing environment to improve performance.

Plan Based on Concurrently Audited Users

When planning, always focus on the number of concurrently audited users, not just the total number of audited systems. Take into consideration user sessions that might be generated as a result of automated monitoring activity from System Monitoring and Management software, such as BMC Patrol etc.

Avoid Single Box Deployment

Always avoid installing key components such as SQL Server, collectors and Audit Management Server on the same system, especially in environments with heavy workload. Keep in mind that a collector's workload is CPU intensive and SQL Server's workload is CPU, IO, and memory intensive. If both a collector and SQL Server are installed on the same system, they'll slow each other down.

Control the Amount of Data

It's always a good practice to establish rules to avoid capturing unnecessary data. This typically includes blacklisting commands such as top or tail (which generate large outputs and seldom contain any meaningful user activity) or enable per-command auditing instead of session auditing. Also, compile a list of users that do not really need to be audited and add them to the non-audited user's list. This often includes user accounts that are used to run automated jobs from System Monitoring and Management software, such as BMC Patrol and so forth.

Scope the Audit Stores Efficiently

Always visualize the flow of traffic, not just when audited activity is being captured but also when it's being searched and replayed. It's better to avoid traffic over slow links by splitting the audited systems into multiple audit stores based on their geographic location, even if it may mean that you'll be deploying more collectors and SQL Servers. In certain cases, splitting audited systems into multiple audit stores may not be sufficient enough and you may even need to consider provisioning multiple audit and monitoring service installations. When audited data is being queried, all calls are routed to the audit store databases by way of the Management database. If the Management database is not connected to the console or to the audit store databases by way of a fast network link, the queries will always return the results slowly no matter how good the performance of SQL Server is.

Estimate Storage Requirement based on Pilot Data

No two customers are the same and you can never accurately predict how much data will be collected over a period of time in each environment. Hence, it's important to analyze existing data in a customer's environment (from pilot project) to predict the future data growth. A pilot testing is an effective way to help you understand a number of things such as the following factors:

- Understand workload patterns and come up with an overall configuration strategy that determines how the audit stores will be scoped, which users should or should not be audited, which commands should be blacklisted and so forth.
- Database storage requirement – Roughly, how much data will be collected over the retention policy period? This will also help you establish the active audit store database rotation policy.
- What kind of hardware will be needed for the SQL Server to serve the production workload?
- How many collectors will be needed in each audit store (this number is especially important when auditing Windows systems)?

The Centrify Audit & Monitoring Service Data Analysis tool (see KB-4496) can be very helpful to understand data trends. If the Centrify Audit & Monitoring Service Data Analysis tool reveals that more than anticipated amount of data is being captured, you can always use the database rotation to keep the active audit store database's size in control thus controlling the storage requirements for all attached databases.

Maintain Databases Periodically

Apart from taking regular backups, it's also important to keep the databases healthy by maintaining them periodically. This includes activities such as reorganizing or rebuilding indexes; these tasks must be done by a customer's DBA periodically. Centrify recommends reorganizing indexes if they are 5% to 30% fragmented and rebuilding indexes if they are more than 30% fragmented.

Control the Size of Active Databases

A large active audit store database often results in poor performance as a result of fragmented indexes, lengthy backups, and out of date database statistics, especially when the databases are not maintained periodically. Centrify recommends keeping the active audit store database size between 250GB-500GB (as of Suite 2016). Consider rotating databases whenever the size exceeds the recommended thresholds. You can rotate databases programmatically by using either the Centrify DirectManage SDK or the Centrify Audit PowerShell Module, or manually using the Audit Manager console). It's also a good practice not to keep too many audit store databases attached to an audit store, because doing so affects query performance.

Plan Database Rotation based on Retention Policy

Always try to align the audit data retention policy with the active audit store database rotation. For example, if the audit data retention policy requires last 90 days of data to be online, try to rotate the active audit store database every 90 days. This strategy makes it easy to find achieved data if it's ever needed for reviewing purpose in the future. One exception to this strategy is an environment where the audit data retention policy is so long that the active audit store database is guaranteed to exceed the recommended maximum size of the active audit store database (as mentioned in the previous section). In such cases, you can divide the entire retention policy period into small periods (for example, one database for each month) and continue to rotate the active audit store database at the recommended intervals. Irrespective of which strategy you choose and implement, it's always recommended to detach all audit store

databases that contain data outside of the retention policy period. This not only improves the query performance but also reduces the disk usage on the database server.

Configure SQL Server Optimally

Centrify recommends setting the SQL Server machine's power plan settings (Control Panel > Power Options) to High Performance.

SQL Server has a setting called Max Server Memory that controls the maximum amount of physical memory that can be consumed by the SQL Server's buffer pool. An incorrectly configured Max Server Memory may either result in the SQL engine causing high IO or OS/other programs starving for more memory. It's critical to configure the Max Server memory correctly based on the amount of total physical memory available. Always configure this value as recommended before deployment begins.

Centrify recommends storing the transaction logs and data files that are associated with any SQL Server database on two separate volumes. For more information, see the Microsoft Knowledge base article https://support.microsoft.com/en-us/kb/2033523.

Other Recommendations

Centrify recommends deploying at least two collectors per audit store for redundancy purpose.

Understand that any Hardware has its Limits

It's entirely possible that even after following all the best practices, the Centrify Audit & Monitoring Service system continues to perform poorly. In such cases, you must consider splitting the workload by deploying additional SQL Servers or collectors, depending on where the bottleneck is. Deploying an additional SQL Server will almost always result in reconfiguring scope of the audit stores (in order to redirect some traffic to the new SQL Server) and it must be done with careful planning.

Creating an initial estimate of your database storage needs

Here's a way that you can do an initial but rough guess of your recommended storage needs. For a more detailed estimate, please refer to Guidelines for determining hardware configuration.

- 1 MB per minute per Windows or Linux Desktop session with nominal activity

  You could need considerably more storage than this if sessions will include flash animation, video replay, and so forth.

- Use a ratio of users to number of systems
  - For example, you could have 50 users for 1000 systems, which would be a 5% ratio; you would then multiply the number of users with the above estimates.
  - 50 Windows users would require 50 MB/minute, or 120 GB/week, or 1.5TB/quarter.

Guidelines for determining hardware configuration

The overall performance of the audit and monitoring service ecosystem ultimately depends on the performance of SQL Server and the collectors. To come up with guidelines for hardware, we have created a test environment wherein the SQL Server hardware configuration has been categorized into three variants: a low end SQL Server, a high end server SQL Server, and a mid-level SQL Server. Below are the test environment configuration details:

Physical machine	DIY PC	S5000 Intel Xeon	Dell R730
Physical memory	8 GB (2x4GB)	16 GB (2x8GB)	32 GB (2x16GB)
CPU	Intel i5-650, 3.2 GHz	E5420 (2.5 GHz)	2xIntel Xeon E5-1620 v3 (2.4 GHz, 8C/16T)
HDD	1x1TB (7200 rpm SATA)	1x1TB (7200 rpm SATA)	1x1TB (7200 rpm SAS 6Gbps)

The hardware configuration depicted in the above table reflects the sizing test environment. Centrify cannot make specific recommendations (such as

physical memory, CPU frequency, or CPU type) for purchasing hardware; use these numbers only as a guideline.

The table below lists the test conditions along with the outcome of tests, and this roughly indicates the recommended number of audited systems that can be supported in this test environment.

Test conditions	60% agents are idle 35% agents are running simple commands	5% agents are running tail command	5% agents are idle 2% agents are running "su" sessions 93% agents are running "dzdo" command sessions	60% agents are idle 40% agents are active	100% agents are active
Low end SQL Server	1100	1800	400	1300	
Mid-range SQL Server	1500	3600	400	2400	
High end SQL Server	2000	4500	640	3000	

- The numbers depicted in the above table reflects the outcome of a sizing test in a very specific test; use these numbers only as a guideline.
- Refer to the table in the next section for actual recommendations.

Based on these test results, Centrify recommends using the table below when planning a deployment of Centrify Audit & Monitoring Service. Please note that the recommended SQL Server configuration is only applicable to the SQL Server hosting the audit store database. It's generally a good practice to host the Management database on the same SQL Server where the other audit store databases are hosted.

UNIX	Command auditing	1800	5% agents are idle 2% agents are running "su" sessions 93% agents are running "dzdo" command sessions	Low end	2	83	
UNIX	Command auditing	3600	5% agents are idle 2% agents are running "su" sessions 93% agents are running "dzdo" command sessions	Mid-range	2	60	
UNIX	Command auditing	4500	5% agents are idle 2% agents are running "su" sessions 93% agents are running "dzdo" command sessions	High end	4	102	
UNIX	Session auditing	1100	60% agents are idle 35% agents are running simple commands 5% agents are running tail command	Low end	2	87	
UNIX	Session auditing	1500	60% agents are idle 35% agents are running simple commands 5% agents are running tail command	Mid-range	2	76	

UNIX	Session auditing	2000	60% agents are idle 35% agents are running simple commands 5% agents are running tail command	High end	4	104	
Windows	Video	disabled	1300	100% agents are active	Low end	2	91
Windows	Video disabled	2400	100% agents are active	Mid-range	3	67	
Windows	Video	disabled	3000	100% agents are active	High end	4	100
Windows or Linux Desktop	Video enabled	400	60% agents are idle 40% agents are active	Low end	5	85	
Windows or Linux Desktop	Video enabled	400	60% agents are idle 40% agents are active	Mid-range	5	88	
Windows or Linux Desktop	Video enabled	640	60% agents are idle 40% agents are active	High end	8	113	

- Expected activity is based on 8 hours of work every day. Results may vary if the target environment has a different pattern for user activity/behavior, different workload/ratio of idle to active systems compared to the test environment.
- Average response time is the total time taken in milliseconds to send a unit of data from audited system to the SQL Server by way of collector.
- All recommended numbers are based on the assumption that the target environment is stable in terms of performance of individual components and network throughput. Intermittent transient errors are expected and typically do not impact the sizing assessments.
- Windows and Linux Desktop audited systems generate large amount of audit data when video capture is enabled and such environments require high performance SQL Server storage. This is the primary reason why the number of agents supported between the low and medium SQL Server configuration are similar. The artificial load generated by the test simulators is also higher than the expected daily activity in a typical production environment. With high performance storage, the total number of Windows and Linux Desktop audited systems supported will likely be higher compared to the numbers recommended.
- When monitoring both Windows and UNIX/Linux audited systems in the same environment, use the Windows numbers as a guideline.

Identifying Typical Deployment Issues

It's fairly easy to identify scaling/performance issues with a Centrify Audit & Monitoring Service system that are typically a result of poor planning or deployment. Below are some of the most common deployment issues.

Large Spool Files on Audited Systems

A healthy audit and monitoring service system should be able to keep up the pace with users' audited activity. When the system cannot keep up the pace, it means either the user's audited activity is generating too much data (such as when a user runs the cat command on a very large file) or the audit and monitoring service system components (such as collectors and databases) are not able to process and store the generated data fast enough. In such cases, you'll typically see large spool files on the audited systems that often need more time to get despooled completely.

Constant High CPU on Collector/SQL Server

It's perfectly normal to see high CPU activity on collector and SQL Server machines during peak hours as this is the time when data is continuously getting pumped from the audited system to the collector and finally to the database. However, when you see similar activity during off-peak hours (especially when it doesn't correspond to the number of active users in that environment at that time), it indicates that the audit and monitoring service system is getting

backlogged.

Low Despool Rate

The despool rate largely depends on the type of data being captured, the speed of network/latency between audited system and collector, the speed of the network/latency between the collector and the database, and ultimately the performance of the SQL Server itself. Because of these factors, there's no ideal value or range for the despool rate. However, you should not see a despool rate that's significantly lower than the rate of data capture, especially when there are no known issues related to network speed or SQL Server performance.

False "Agent disconnected" Alerts

Each Agent periodically sends its heartbeat to the database (by way of collector) and the Audit Manager console relies on this ping to determine if the Agent is connected or not. If there are deployment issues with audit and monitoring service, the Agent heartbeat may not get registered even if the Agent is online, and this may raise false alarms as the system will be shown as disconnected in Audit Manager Console. Whenever you see such contradicting information regarding the status of the audited system, it typically is indicative of underlying deployment issues.

Too many SQL Server Tasks In Queue

SQL Server has a fixed set of worker threads that it can use to perform its job and this number depends on the CPU architecture, such as 32-bit or 64-bit, and the total number of CPUs on the SQL Server. If SQL Server is given more tasks than it can finish, they'll end up waiting at the bottom of this queue, thus consuming memory and degrading overall system performance. Always consult the DBA to confirm if the environment is consistently showing a lot of tasks in the worker queue; this can indicate that the workload is too much for this SQL Server to handle. For more information, see the Microsoft article https://msdn.microsoft.com/en-us/library/ms177526(v=sql.105).aspx.

Settings to Adjust for Performance Improvement

In an environment where Centrify Audit & Monitoring Service is already deployed and experiencing scalability/performance issue, it's not always possible to re-architect the deployment or make significant configuration changes (such as re-scoping the audit stores or adding a new SQL Server may not be practical); this is true especially in large environments. The table below that lists some key settings that you may try to change in order to improve the overall performance of various audit and monitoring service components.

Agent Settings			
Agent heartbeat interval for Unix/Linux Audited Systems (dad.timer.update.agent.status)	Controls the interval for sending Unix/Linux Audited System's heartbeat to the Collector	When SQL activity monitor shows high CPU usage at a predictable interval as a result of heartbeat registration or when Collector logs reveal repeated failures in registering Audited System's heartbeat or when Audit Manager console frequently shows a lot of disconnected Audited Systems even if they are all online. For more details about the configuration parameter, see the Configuration and Tuning Reference Guide.	Unix/Linux Agent (centrifyda.conf)
Agent heartbeat interval for Windows Audited Systems (SessionPingInterval)	Controls the interval for sending Windows Audited System's heartbeat to the Collector	When the SQL activity monitor shows high CPU usage at a predictable interval as a result of heartbeat registration or when Collector logs reveal repeated failures in registering Audited System's heartbeat or when Audit Manager console frequently shows a lot of disconnected Audited Systems even if they are all online.	Windows Agent (registry setting)
User blacklisting (dash.user.skiplist)	Allows specifying blacklist of users that should not be audited on Unix/Linux systems	Useful in preventing capture of audit activity of users such as BMC Patrol agent or ServiceNow service accounts or users that do not really need to be audited. For more details about the configuration parameter, see the Configuration and Tuning Reference Guide.	Unix/Linux Agent (centrifyda) and also available via Group Policy
Audited/Non-audited users list	Allows specifying whitelist or blacklist of users that should or should not be audited on Windows systems	Useful in preventing capture of audit activity of unwanted users.	Group Policy

BindingCheckInterval	Controls the interval at which Agent checks if it's connected to the correct Collector or not	When binding check causes load on the Domain Controller as a result of periodic Active Directory calls (for example, when you notice an Active Directory call from each Audited System every 10 seconds)	Windows Agent (registry setting)
Collector settings			
Agent global heartbeat interval (AgentMinimumUpdateInterval)	Controls the interval for sending Audited System's heartbeat to the Collector at the Collector level (in case it's not practical to tweak this setting on each of the Audited Systems)	When SQL activity monitor shows high CPU usage at a predictable interval as a result of heartbeat registration or when Collector logs reveal repeated failures in registering Audited System's heartbeat or when Audit Manager console frequently shows a lot of disconnected Audited Systems even if they are all online.	Collector (registry setting)
Maximum concurrent SQL connections per Collector (MaxPoolSize)	Controls how many SQL connections (maximum) can be opened by the Collector at a time	In order to reduce the workload caused by Collector on the SQL Server. Reducing the MaxPoolSize will reduce the total number of connections open on the SQL Server but may also reduce the despool rate.	Collector (registry setting)
Installation level settings			
Command blacklisting	Allows specifying one or more commands whose output is not required to be captured	When you see large audited sessions that are a result of running commands with large output (for example, commands such as tail or top) and you need to control disk space consumed by such audited activity.	Group Policy
Enable/Disable video audit	Allows enabling or disabling video capture (at installation level or on a per machine basis) when storing audited user activity in the database	When video capture is resulting into large sessions consuming a lot of disk space and/or it's not desirable to store the video.	Audit Manager console or group policy

- Not all configuration parameters/settings are available in releases prior to Suite 2015.1. Please contact Centrify Support for additional information on older releases.
- Agent heartbeat interval can configured per audited system or globally by configuring it in collector's registry setting. Centrify recommends configuration the heartbeat interval on the collector if you want all the audited systems to send their heartbeat at an identical interval.
- Tweaking the configuration settings may not always help or eliminate the deployment issues completely. In such cases, making significant deployment/configuration changes may be the only option. Please contact Centrify Support to evaluate possible solutions.

Conclusion

This sizing recommendation section has provided some information as to what factors can affect Centrify Audit & Monitoring Service performance. Keep in mind, however, that every installation is unique and we cannot anticipate every use case. If you continue seeing performance degradation after following the best practices outlined in this document, contact Centrify Support for assistance.

Server Suite provides an auditing infrastructure that enables your organization to capture and store session activity on audited computers. The auditing infrastructure also enables auditors to query and report on specific events, view all or selected session activity, change the status of reviewed sessions, and delete sessions that are no longer needed. The auditing infrastructure relies on two types of databases to store information: the management database and the audit store database.

If you are not familiar with the components and architecture of the auditing infrastructure, see the Auditing Administrator's Guide. That guide provides detailed information about how the components in the auditing infrastructure communicate with each other and how to configure and manage an audit installation.

Introduction to the Databases Used for Auditing

Server Suite provides an auditing infrastructure that enables your organization to capture and store session activity on audited computers. The auditing infrastructure also enables auditors to query and report on specific events, view all or selected session activity, change the status of reviewed sessions, and delete sessions that are no longer needed. The auditing infrastructure relies on two types of databases to store information: the **management database** and the **audit store database**.

If you are not familiar with the components and architecture of the auditing infrastructure, see the *Auditing Administrator's Guide*. That guide provides detailed information about how the components in the auditing infrastructure communicate with each other and how to configure and manage an audit installation.

Management Databases Store Installation Information

In most organizations, there is only one management database for each audit installation and it stores information about the components of the auditing infrastructure for that installation. For example, the management database stores information about which computers are audited, where the collector service is installed, and the scope (site or subnet) of each audit store.

In most cases, you create the management database when you create a new installation and update it whenever you add components to the auditing infrastructure. For example, if you enable auditing on additional computers or deploy the collector service on a new server, the change is recorded in the management database. Because the management database stores information about the auditing infrastructure and not audited sessions, it requires little to no maintenance over time.

Audit Store Databases Store Audited Sessions

Like the management database, you create the first audit store database during deployment. However, unlike the management database, the audit store database stores the activity collected from audited computers. Over time, the audit store database would grow and become unmanageable. Therefore, most organizations periodically add a new audit store database to capture current activity. When the new audit store database becomes active, the previous audit store database can remain “attached” to provide access to stored information or be “detached” if access to the information stored in that database is no longer required.

The process of adding a new audit store database and changing the status of an existing audit store database from “active” to “attached” is called database rotation. Database rotation is the primary on-going administrative task to manage the auditing of user activity using Centrify software. There are, however, also steps to take during the planning phase and during deployment that apply specifically to preparing Microsoft SQL Server to support the auditing infrastructure.

The audit store database stores all of the activity collected on audited computers. When auditors or administrators want to review captured activity, they must be able to connect to the audit store database to retrieve it. Therefore, the audit store database must be accessible and the auditors and administrators who need to retrieve data from it must have the appropriate permissions to connect to the database instance, and to read and write data where applicable.

Using Multiple Databases for the Audit Store

Depending on the number of computers you are auditing, the level of detail you capture, and the length of time captured activity must be available for review, an audit store database can grow too large to manage effectively in a short period of time.

To prevent the audit store database from growing too large, you can split it into multiple databases. Only one database at a time can be the active—that is, the database currently receiving captured activity from an audited computer and its collector service.

However, because large databases are harder to manage and take longer to search than smaller ones and you cannot allow a single active database to grow indefinitely, you can change the active database to be an attached database—that is, available for searching and retrieving stored information but no longer receiving captured activity—and make a new database the currently active database.

Changing which database is active without interrupting the monitoring of audited computers is also referred to as *rolling* or *rotating* the database. By adding new databases and changing the audit store's active database to an attached database before it gets too large, you can optimize database performance and storage requirements.

Detaching and Retiring Audit Store Databases

All of the information stored in audit store databases that are attached to an audit installation is available for queries and reports and can be viewed in the session player. When the information in an attached database is no longer needed, you can detach the database from the installation. You cannot detach a database while it is the active database.

After a database that has been the active database is made an attached or detached database, it is considered a retired or decommissioned database. It cannot be used as the active database again.

Automating Database Rotation

Although you can do database rotation manually using Audit Manager, you might want to automate the process to perform it automatically on a regular schedule. You might also want to automate and schedule the detachment of old databases from the audit store. The API described in the reference enables you to write scripts to perform database rotation and attach or detach databases.

The software development kit (SDK) for auditing includes four sample scripts that you can modify to suit your purposes: two VBScript samples and two Power Shell samples. One pair of sample scripts (db_rotation) use default database settings. The second pair (db_rotation_sql_script) let you customize the database scripts to set up the database and the server.

The sample scripts perform the following steps:

1. Create a new audit store database and attach it to an audit store.
2. Grant permission to the management database and collectors to access the newly created audit store database.
3. Make the newly created audit store database the active database.
4. Detach any audit store databases older than two years.
5. Publish the settings to Active Directory so that audited computers and collectors can look up the information.

Note that the sample scripts require the user to respond to informational messages at various points during execution. To make these scripts run without user interaction, remove or comment out all the wscript.echo commands in the script, or redirect the echo commands to STDOUT so that the scheduled task will not hang waiting for user input.

The following command adds the script db_rotation.vbs as a monthly scheduled task named rolldb to be run as user domain_name\administrator. By using cscript.exe to launch the script, it redirects output to STDOUT.

PS C:\Program Files\Centrify\Audit\SDK\Samples> schtasks.exe /Create /TN "rolldb" /TR "cscript.exe 'C:\Program Files\Centrify\Audit\SDK\Samples\db_rotation.vbs' DefaultInstallation DefaultAuditStore sqlserver.domain_name.com subtest3" /RU domain_name\administrator /SC Monthly /MO 1

The components of this command are as follows:

Schtasks.exe /Create /TN <*Task_name*> /TR <*Task_Command*> /RU <*Run_As_User*> /SC <*Reoccurance_rate*> /MO <*Reoccurance_increment*>

where

- *Task_Name*: rolldb
- *Task_Command*: cscript.exe 'C:\Program Files\Centrify\Audit\SDK\Samples\db_rotation.vbs' DefaultInstallation DefaultAuditStore sqlserver.domain_name.com subtest3
- *Run_as_user*: domain_name\Administrator
- *Reoccurance_rate*: Monthly
- *Reoccurance_increment*:1

The task command consists of the following elements:

<*parser*> '<*install_path*>\<*VBS_script*>' <*Installation*> <*auditstore*> <*DB_Server*> <*DB_prefix*>

where

- *parser*: cscript.exe
- *install_path*: C:\Program Files\Centrify\Audit\SDK\Samples
- *VBS_script*: db_rotation.vbs
- *Installation*: DefaultInstallation
- *auditstore*: DefaultAuditStore
- *DB_Server*: sqlserver.domain_name.com

- *DB_prefix*: subtest3

The prefix is attached to a date stamp in the name of the newly created audit store database.

Delinea

Audit-related object reference

This chapter describes the classes, methods, and properties in the Centrify software development kit for auditing. The following classes are used for managing auditing features and are defined in the Centrify.DirectAudit.API namespace:

Account class	Manages Account objects.
Accounts class	Enumerates Account objects.
AuditServer class	Manages AuditServer objects.
AuditServers class	Enumerates AuditServer objects.
AuditStore class	Manages AuditStore objects.
AuditStoreDatabase class	Manages AuditStoreDatabase objects.
AuditStoreDatabases class	Enumerates AuditStoreDatabase objects.
Connection class	Manages an auditing connection.
Installation class	Manages Installation objects.

Account Class

Manages Account objects.

Syntax

class Account

Properties of the Account class

The Account class provides the following properties:

IsSystemAccount property	Gets a value indicating whether the account is a Windows system account.
IsWindowsAccount property	Gets a value indicating whether the account is a Windows domain account.
UserName property	Gets the user name of the account.

Description of the Account class

The accounts used for auditing include the management database account, audit store database account, and collector accounts. This class provides properties to retrieve information about an account.

See also

- Accounts class
- OutgoingAccount property
- AuditServerAccounts property
- CollectorAccounts property

IsSystemAccount Property

Gets a value indicating whether the account is a Windows system account.

Syntax

bool IsSystemAccount{get;}

Return Value

Returns true if the account is a Windows system account; otherwise, false.

Discussion of the IsSystemAccount Property

When you attach a new database to the audit store, you must set the database to allow access by the management database account. Before you call the AddAuditServerAccount method, you should check to see if the management database account is a Windows system account because if it is, the Account.UserName property is not a Windows domain account name and therefore cannot be passed directly to the AddAuditServerAccount method.

Example

The following code sample first checks to make sure the management database account is not a system account. If it is not a system account, the sample calls the AddAuditServerAccount method. If the management database is a system account, the sample returns an error message.

...

```
' Grant permission to management database to access the audit store database
SET objAuditServers = objInstallation.AuditServers
FOR EACH objAuditServer IN objAuditServers
SET objAuditServerAccount = objAuditServer.OutgoingAccount
IF NOT objAuditServerAccount.IsSystemAccount THEN
objAuditStoreDatabase.AddAuditServerAccount objAuditServerAccount.UserName, & _

objAuditServerAccount.IsWindowsAccount
wscript.echo "Added management database account '" & objAuditServerAccount.UserName & "'."
ELSE
wscript.echo "Cannot add account for management database '" & objAuditServer.Name & _
& "' because the account '" & objAuditServerAccount.UserName & _
& "' is a system account."
wscript.echo "NOTE: Please add allowed incoming management database for '" & _

& objAuditServer.Name & _
& "' to the new audit store database in Audit Manager."
END IF
```

See also

- IsWindowsAccount property
- AddAuditServerAccount method

IsWindowsAccount Property

Gets a value indicating whether the account is a Windows domain account.

Syntax

```
bool IsWindowsAccount {get;}
```

Return Value

Returns true if the account is a Windows domain account; false if the account is an SQL Server login account.

Discussion

The management database-to-audit store database connection and the collector-to-audit store connection can use either Windows authentication or SQL Server authentication.

Example

The following code sample illustrates using this property as an input parameter to the AddAuditServerAccount method:

...

```
'Add management database accounts for those management databases running in
' system account; e.g. NT Authority/Network Service
'
DIM strAuditServerAccount
DIM isAuditServerWindowsAccount
isAuditServerWindowsAccount = true
strAuditServerAccount = "DOMAIN\MACHINE$"
objAuditStoreDatabase.AddAuditServerAccount strAuditServerAccount, & _
isAuditServerWindowsAccount
wscript.echo "Added management database account '" & strAuditServerAccount & "'."
```

See also

- AddAuditServerAccount method
- AddCollectorAccount method

UserName Property

Gets the user name of the account.

Syntax

string UserName {get;}

Return Value

Returns the user name of the account.

Discussion

If the account is a Windows account, the user name is the Windows domain account name. If the account is an SQL Server login account, the user name is the SQL Server account name.

Example

The following code sample illustrates using this property as an input parameter to the AddCollectorAccount method:

...

```
' Copy Collector accounts from current active audit store database
SET objCollectorAccounts = objActiveDatabase.CollectorAccounts
FOR EACH objCollectorAccount IN objCollectorAccounts
objAuditStoreDatabase.AddCollectorAccount objCollectorAccount.UserName
wscript.echo "Added Collector account '" & objCollectorAccount.UserName & "'."
```

Accounts class

Enumerates Account objects.

Syntax

class Accounts

Discussion

The accounts used for auditing include the management database account, the audit store database account, and collector accounts. Use this class to enumerate a set of accounts.

Example

In the following code sample, the CollectorAccounts property returns an Accounts object and a FOR EACH—IN statement is used to enumerate the collector accounts:

...

```
' Copy Collector accounts from current active audit store database
SET objCollectorAccounts = objActiveDatabase.CollectorAccounts
FOR EACH objCollectorAccount IN objCollectorAccounts
objAuditStoreDatabase.AddCollectorAccount objCollectorAccount.UserName
wscript.echo "Added Collector account '" & objCollectorAccount.UserName & "'."
```

See also

- Account class
- AuditServerAccounts property
- CollectorAccounts property

AuditServer class

Manages AuditServer objects.

Syntax

class AuditServer

Properties

The AuditServer class provides the following properties:

DatabaseName property	Gets the database name of the management database.
Name property (management database)	Gets the display name of the management database.
OutgoingAccount property	Gets the outgoing account of the management database.
ServerName property	Gets the Microsoft SQL Server instance name of the management database.

Discussion

An AuditServer object holds information about an management database that is part of the audit installation. The management database stores license information, audit roles, and information about the components of the auditing infrastructure, including the scope of each audit store and the active and attached audit store databases.

See also

- AuditServers class

DatabaseName Property

Gets the database name of the management database.

Syntax

string DatabaseName {get;}

Return Value

Returns the database name of the management database.

Discussion

The management database stores license information, audit roles, and information about the components of the auditing infrastructure, including the scope of each audit store and the active and attached audit store databases.

See also

- Name property (management database)
- ServerName property
- AuditStoreDatabase class

Name Property (management database)

Gets the display name of the management database.

Syntax

string Name {get;}

Return Value

Returns the display name of the management database.

Discussion

The management database display name is used in the Audit Manager console and must be unique in the installation. Note that this is not the management database instance name, which is the fully-qualified domain name of the management database, and is not necessarily the same as the management database name, which need not be unique in the installation.

See also

- DatabaseName property
- ServerName property

OutgoingAccount Property

Gets the outgoing account of the management database.

Syntax

Account class OutgoingAccount {get;}

Return Value

Returns the outgoing account of the management database.

Discussion

The user name of the outgoing account is the name by which the management database identifies itself when connecting to an audit store.

ServerName Property

Gets the Microsoft SQL Server instance name of the management database.

Syntax

string ServerName {get;}

Return Value

Returns the Microsoft SQL Server instance name of the management database.

Discussion

The Microsoft SQL Server instance name is the fully qualified domain name of the management database.

See also

- DatabaseName property
- Name property (management database)

AuditServers class

Enumerates AuditServer objects.

Syntax

class AuditServers

Discussion

In most cases, an audit installation includes only one management database.

See also

- AuditServer class

AuditStore class

Manages AuditStore objects.

Syntax

class AuditStore

Properties

The AuditStore class provides the following properties:

ActiveDatabase property	Gets the active audit store database.
Databases property	Gets the list of the audit store databases.
Name property (audit store)	Gets the display name of the audit store.

Methods

The AuditStore class provides the following methods:

AddDatabase method	Creates a new audit store database and attaches the database to the audit store using default settings.
AddDatabaseByScript method	Creates a new audit store database and attaches the database to the audit store using custom settings specified in SQL scripts.
AttachDatabase method	Attaches an existing audit store database to the audit store.
ChangeActiveDatabase method	Changes which database is currently active in the audit store.
DetachDatabase method	Detaches a database from the audit store.
GetDatabase method	Retrieves the audit store database object given the database display name.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. This class allows you to manage the audit store, including attaching and detaching databases and specifying which database is active. To get information about the attached databases, use the AuditStoreDatabase class and the AuditStoreDatabases class.

See also

- AuditStoreDatabase class

ActiveDatabase Property

Gets the active audit store database.

Syntax

AuditStoreDatabase class ActiveDatabase {get;}

Return Value

Returns the active audit store database.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time.

See also

- Databases property

Databases Property

Gets the list of the audit store databases.

Syntax

AuditStoreDatabases class Databases {get;}

Return Value

Returns the list of the audit store databases.

Discussion

This property returns a list of all the databases attached to the audit store.

See also

- ActiveDatabase property

Name property (audit store)

Gets the display name of the audit store.

Syntax

string Name {get;}

Return Value

Returns the display name of the audit store.

Discussion

The display name of the audit store is used in the Audit Manager console. It is distinct from the display name of the active database.

See also

- Name property (audit store database)

AddDatabase Method

Creates a new audit store database and attaches the database to the audit store using default settings.

Syntax

AuditStoreDatabase class AddDatabase(

string *name*,

string *serverName*,

string *database*

)

Parameters

Return Value

Returns the AuditStoreDatabase object of the new audit store database.

Errors

The AddDatabase method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage Database permission on the audit store or you do not have the SQL Server permission to create SQL Server databases on the Microsoft SQL Server instance.
- Centrify.DirectAudit.Common.Logic.AlreadyExistsException if the specified display name is already being used by another audit store database, or the specified database name is already being used by another database in the Microsoft SQL Server instance.

Discussion

Use this method to create a new audit store database and attach it to the audit store. To customize the database or attach an existing database to the audit store, use one of the methods listed in the “See also” section.

Example

The following code sample argument illustrates the use of AuditStore.AddDatabase:

```
...

strInstallationName = wscript.arguments.item(0)
strAuditStoreName = wscript.arguments.item(1)
strServerName = wscript.arguments.item(2)
strDatabaseName = wscript.arguments.item(3)

SET objAuditStoreDatabase = objAuditStore.GetDatabase(strDatabaseName)

IF NOT objAuditStoreDatabase IS NOTHING THEN
wscript.echo "Audit Store database '" & strDatabaseName & "' already exists."
wscript.quit
END IF

' Create a new audit store database and attach to the audit store
SET objAuditStoreDatabase = objAuditStore.AddDatabase(strDatabaseName, & _
strServerName, strDatabaseName)

IF objAuditStoreDatabase IS NOTHING THEN
wscript.echo "Failed to add audit store database '" & strDatabaseName & "'."
wscript.quit
END IF
wscript.echo "Created and attached audit store database '" & strDatabaseName & "'."
```

See also

- AddDatabaseByScript method
- AttachDatabase method
- ChangeActiveDatabase method

AddDatabaseByScript Method

Creates a new audit store database and attaches the database to the audit store using custom settings specified in SQL scripts.

Syntax

AuditStoreDatabase class AddDatabaseByScript(

string *name*,

string *serverName*,

string *database*,

string *scriptFile1*,

string *scriptFile2*

)

Parameters

Return Value

Returns the AuditStoreDatabase object of the new audit store database.

Errors

The AddDatabaseByScript method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage Database permission on the audit store or you do not have the SQL Server permission to create SQL Server databases on the Microsoft SQL Server instance.
- Centrify.DirectAudit.Common.Logic.AlreadyExistsException if the specified display name is already being used by another audit store database, or the specified database name is already being use by another database in the Microsoft SQL Server instance.

Discussion

The database name you specify in the database parameter is substituted for the keyword #database in the SQL script. To create a new database using standard settings or to attach an existing database to the audit store, use on the methods listed in the "See also" section.

Example

The following code sample illustrates using AuditStore.AddDatabaseByScript in a script:

...

' Create a new audit store database and attach to the audit store

```
SET objAuditStoreDatabase = objAuditStore.AddDatabaseByScript(strDatabaseName, & _
strServerName, strDatabaseName, strServerScriptFile, strDatabaseScriptFile)

IF objAuditStoreDatabase IS NOTHING THEN
wscript.echo "Failed to add audit store database '" & strDatabaseName & "'."
wscript.quit
END IF
wscript.echo "Created and attached audit store database '" & strDatabaseName & "'."
```

See also

- AddDatabase method
- AttachDatabase method
- ChangeActiveDatabase method

AttachDatabase method

Attaches an existing audit store database to the audit store.

Syntax

AuditStoreDatabase class AttachDatabase(

string *name*,

string *server*,

string *database*

)

Parameters

Return Value

Returns the AuditStoreDatabase object of the attached audit store database.

Errors

The AttachDatabase method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage Database permission on the audit store or you do not have the SQL Server permission to create SQL Server databases on the Microsoft SQL Server instance.
- Centrify.DirectAudit.Common.Logic.AlreadyExistsException if the specified display name is already being used by another audit store database, or the specified database name is already being use by another database in the Microsoft SQL Server instance.

Discussion

Use this method if you already have a database that you want to attach to the audit store. To create a new database and attach it to the audit store, use the AddDatabase or AddDatabaseByScript method instead.

Example

The following code sample illustrates using AuditStore.AttachDatabase in a script:

...

```
' Attach an audit store database to the audit store
SET objAuditStoreDatabase = objAuditStore.AttachDatabase(strDatabaseName, & _
strServerName, strDatabaseName)

IF objAuditStoreDatabase IS NOTHING THEN
wscript.echo "Failed to attach audit store database '" & strDatabaseName & "'."

wscript.quit
END IF
wscript.echo "Attached audit store database '" & strDatabaseName & "'."
```

See also

- AddDatabase method
- AddDatabaseByScript method

ChangeActiveDatabase Method

Changes which database is currently active in the audit store.

Syntax

void ChangeActiveDatabase(

AuditStoreDatabase class*database*

)

Parameters

Errors

The ChangeActiveDatabase method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage Database permission on the audit store or you do not have the SQL Server permission to create SQL Server databases on the Microsoft SQL Server instance.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. Once you have made a database inactive by calling this method, you cannot make it active again. You cannot detach the active database.

Example

The following code sample illustrates using AuditStore.ChangeActiveDatabase in a script:

```
' Change active Audit Store database
objAuditStore.ChangeActiveDatabase(objAuditStoreDatabase)
wscript.echo "Changed active database to '" & objAuditStore.ActiveDatabase.Name & "'."
```

See also

- IsActive property

DetachDatabase Method

Detaches a database from the audit store.

Syntax

void DetachDatabase(

AuditStoreDatabase class*database*

)

Parameters

Errors

The DetachDatabase method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage Database permission on the audit store or you do not have the SQL Server permission to create SQL Server databases on the Microsoft SQL Server instance.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. You cannot detach the active database.

Example

The following code sample illustrates using AuditStore.DetachDatabase in a script:

...

```
' Detach any Audit Store databases older than 2 years
FOR EACH objDatabase IN objAuditStore.Databases
IF DateDiff("d", today, objDatabase.ActiveEndTime) > 728 THEN
objAuditStore.DetachDatabase(objDatabase)
wscript.echo "Detached Audit Store database '" & objDatabase.Name & "'."
END IF
```

GetDatabase Method

Retrieves the audit store database object given the database display name.

Syntax

AuditStoreDatabase class GetDatabase(

string *displayname*

)

Parameters

Return Value

Returns the AuditStoreDatabase object of the specified database.

Errors

The GetDatabase method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance that hosts the management database or you do not have permission to connect to the Microsoft SQL Server instance of the audit store database to be created.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the instance is not running or does not allow remote connections.

Discussion

Use this method to obtain the audit store database object of any database attached to the audit store if you already have the audit store database display name.

Example

The following code sample illustrates using AuditStore.GetDatabase in a script:

```
...

today = Date
strDatabaseName = strDatabaseName & "-" & Year(today) & "-" & Month(today) & _

& "-" & Day(today)

SET objAuditStoreDatabase = objAuditStore.GetDatabase(strDatabaseName)

IF NOT objAuditStoreDatabase IS NOTHING THEN
wscript.echo "Audit Store database '" & strDatabaseName & "' already exists."
wscript.quit
END IF
```

AuditStoreDatabase Class

Manages AuditStoreDatabase objects.

Syntax

class AuditStoreDatabase

Properties

The AuditStoreDatabase class provides the following properties:

ActiveEndTime property	Gets the end time of a formerly active database.
ActiveStartTime property	Gets the start time of an active or formerly active database.
AuditServerAccounts property	Gets the list of management database accounts that are allowed to access this audit store.
CollectorAccounts property	Gets the list of collector accounts that are allowed to access this audit store.
DatabaseName property	Gets the audit store database name.
IsActive property	Indicates whether this database is the current active database in the audit store.
IsRetired property	Indicates whether this database was formerly the active database and is now retired.
Name property (audit store database)	Gets the display name of the audit store database.
ServerName property	Gets the Microsoft SQL Server instance name of the audit store database.

Methods

The AuditStoreDatabase class provides the following methods:

AddAuditServerAccount method	Adds a management database account to the list of accounts allowed to access this audit store database.
AddCollectorAccount method	Adds a collector account to the list of accounts allowed to access this audit store database.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. This class provides information about any attached database. You can also add an management database or collectors to the list of accounts allowed access to an audit store database. To get information about the audit store, use the AuditStore class.

See also

- AuditStoreDatabases class
- AuditStore class

ActiveEndTime Property

Gets the end time of a formerly active database.

Syntax

DateTime ActiveEndTime {get;}

Return Value

Returns the end time of the database's active period. If the database was never active or is currently active, the return value is System.DateTime.MinValue (12:00:00 AM).

See also

- ActiveStartTime property
- IsRetired property

ActiveStartTime Property

Gets the start time of an active or formerly active database.

Syntax

DateTime ActiveStartTime {get;}

Return Value

Returns the start time of the database's active period. If the database was never active, the return value is System.DateTime.MinValue (12:00:00 AM).

See also

- ActiveEndTime property
- IsRetired property

AuditServerAccounts Property

Gets the list of management database accounts that are allowed to access this audit store.

Syntax

Accounts class AuditServerAccounts {get;}

Return Value

Returns the list of allowed incoming management database accounts.

Discussion

Although most audit installations include only one management database, it's possible to add more.

See also

- CollectorAccounts property
- AddAuditServerAccount method

CollectorAccounts Property

Gets the list of collector accounts that are allowed to access this audit store.

Syntax

Accounts class CollectorAccounts {get;}

Return Value

Returns the list of allowed incoming collector accounts.

See also

- AuditServerAccounts property
- AddCollectorAccount method

DatabaseName Property

Gets the audit store database name.

Syntax

string DatabaseName {get;}

Return Value

Returns the database name of the audit store database.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. This property returns the database name of the database.

To get information about the active database attached to the management database, use the AuditServer class.

See also

- Name property (audit store database)
- ServerName property
- DatabaseName property

IsActive Property

Indicates whether this database is the current active database in the audit store.

Syntax

Bool IsActive {get;}

Return Value

Returns true if the database is the current active database in the audit store; otherwise, false.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time.

See also

- ChangeActiveDatabase method
- IsRetired property

IsRetired Property

Indicates whether this database was formerly the active database and is now retired.

Syntax

Bool IsRetired {get;}

Return Value

Returns true if the database was formerly the active database for the audit store and is now retired; otherwise, false.

Discussion

An audit store can have multiple databases attached, but only one can be active at a time. Once a database has been retired, it cannot be made active again.

See also

- ChangeActiveDatabase method
- IsActive property

Name Property (Audit Store Database)

Gets the display name of the audit store database.

Syntax

string Name {get;}

Return Value

The display name of the audit store database.

Discussion

The display name of the audit store database is the name used in the Audit Manager console when displaying information about the database.

Example

...

wscript.echo "Changed active database to '" & objAuditStore.ActiveDatabase.Name & "'."

See also

- DatabaseName property
- ServerName property

ServerName Property

Gets the Microsoft SQL Server instance name of the audit store database.

Syntax

string ServerName {get;}

Return Value

Returns the Microsoft SQL Server instance name of the audit store database.

Discussion

The SQL Server instance name of the audit store database is the fully qualified domain name of the SQL Server to which the audit store database is attached.

See also

- DatabaseName property
- Name property (audit store database)

AddAuditServerAccount Method

Adds a management database account to the list of accounts allowed to access this audit store database.

Syntax

void AddAuditServerAccount(

string *userName*,

bool *isWindowsAccount*

)

Parameters

Errors

The AddAuditServerAccount method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance or the management database.
- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the Microsoft SQL Server instance is not running and does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage SQL Login permission on the audit store.

Discussion

When you attach a new database to the audit store, you must set the database to allow access by the management database account. If the management database account is a Windows system account, you must explicitly specify the Windows domain account name in the username parameter. For other Windows accounts and for SQL accounts, you can pass the management database's Account.UserName property to this method as the user name.

Example

The following code sample first checks each account to see if it's a Windows system account. If the installation does not use a system account, the code passes the Account.UserName property to the AddAuditServerAccount method as the user name. If the installation uses a system account, it passes the Windows domain account name instead.

...

' Grant permission to management database to access the audit store database

SET objAuditServers = objInstallation.AuditServers

FOR EACH objAuditServer IN objAuditServers

SET objAuditServerAccount = objAuditServer.OutgoingAccount

IF NOT objAuditServerAccount.IsSystemAccount THEN

```
objAuditStoreDatabase.AddAuditServerAccount & _
objAuditServerAccount.UserName, & _
objAuditServerAccount.IsWindowsAccount
wscript.echo "Added management database account '" & objAuditServerAccount.UserName & "'."
ELSE
'Add management database accounts for those management databases running in
' system account; e.g. NT Authority/Network Service
'
DIM strAuditServerAccount
DIM isAuditServerWindowsAccount
isAuditServerWindowsAccount = true
strAuditServerAccount = "DOMAIN\MACHINE$"
objAuditStoreDatabase.AddAuditServerAccount strAuditServerAccount, & _
isAuditServerWindowsAccount
wscript.echo "Added management database account '" & strAuditServerAccount & "'."
END IF
NEXT
```

See also

- AddCollectorAccount method
- AuditServerAccounts property
- IsSystemAccount property
- IsWindowsAccount property
- UserName property

AddCollectorAccount Method

Adds a collector account to the list of accounts allowed to access this audit store database.

Syntax

```
void AddCollectorAccount(
string userName,
)
```

Parameters

Errors

The AddCollectorAccount method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance or the

management database.

- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the Microsoft SQL Server instance is not running and does not allow remote connections.
- Centrify.DirectAudit.Common.Logic.UnauthorizedException if you do not have the Manage SQL Login permission on the audit store.

Discussion

When you attach a new database to the audit store, you must set the database to allow access by each collector account that passes data to that audit store. You can pass the collector's Account.UserName property to this method as the user name.

Example

The following code sample illustrates using AuditStoreDatabase.AddCollectorAccount in a script:

...

```
' Copy Collector accounts from current active Audit Store database
SET objCollectorAccounts = objActiveDatabase.CollectorAccounts
FOR EACH objCollectorAccount IN objCollectorAccounts
objAuditStoreDatabase.AddCollectorAccount objCollectorAccount.UserName
wscript.echo "Added Collector account '" & objCollectorAccount.UserName & "'."
NEXT
```

See also

- AddAuditServerAccount method
- CollectorAccounts property

AuditStoreDatabases class

Enumerates AuditStoreDatabase objects.

Syntax

class AuditStoreDatabases

Example

In the following code sample, the AuditStore.Databases property returns an AuditStoreDatabases object and a FOR EACH—IN statement is used to enumerate the audit store databases:

...

```
' Detach any Audit Store databases older than 2 years
FOR EACH objDatabase IN objAuditStore.Databases
IF DateDiff("d", today, objDatabase.ActiveEndTime) > 728 THEN
objAuditStore.DetachDatabase(objDatabase)
wscript.echo "Detached Audit Store database '" & objDatabase.Name & "'."
END IF
NEXT
```

See also

- AuditStoreDatabase class
- AuditStore class

Connection class

Manages an auditing connection.

Syntax

class Connection

Constructors

The Connection class provides the following overloaded constructor:

Connection constructor	Creates a Connection object.

Methods

The Connection class provides the following overloaded method:

GetInstallation method	Retrieves an audit installation by name or by management database connection.

Discussion

The Active Directory domain controller stores information about the audit installation, including the installation name and the management database being used by the installation. The Connection object provides a way to connect to an Active Directory domain controller and retrieve the installation information stored there.

See also

- Installation class

Connection Constructor

Creates a Connection object.

Syntax

Connection()

Connection(string *domainController*)

Parameters

Specify the following parameter when needed:

domainController	The domain controller of the Active Directory domain to which you wish to connect in order to get information about the audit installation.

Discussion

The Connection object constructor is overloaded. Use the constructor without parameters to create a connection in the current domain. Use the second version of the constructor if you want to specify the Active Directory domain of the connection in order to administer an audit installation on an Active

Directory domain other than the one to which your workstation is joined.

GetInstallation Method

Retrieves an audit installation by name or by management database connection.

Syntax

Installation class GetInstallation(

string *installationName*)

Installation class GetInstallation(

string *server*,

string *database*)

Parameters

Return value

Returns the Installation object found.

Errors

The GetInstallation method may throw the following exception:

- Centrify.Cfw.DirectoryServices.ServerNotOperationalException if the domain controller is not operational. Check to make sure you entered the correct domain name when you called the constructor for the Connection object.

Discussion

The Connection.GetInstallation method is overloaded to provide two ways to search for an installation: by the name of the installation, or by the management database that is part of the installation.

Example

The following code sample illustrates using Connection.GetInstallation in a script to get the Installation object for the audit installation in the current Active Directory domain. The Installation object is then used to get the name of the object store database:

...

```
SET objInstallation = objConnection.GetInstallation(strInstallationName)
SET objAuditStore = objInstallation.GetAuditStore(strAuditStoreName)
SET objAuditStoreDatabase = objAuditStore.GetDatabase(strDatabaseName)
```

See also

- Installation class

Installation class

Manages Installation objects.

Syntax

class Installation

Properties

The Installation class provides the following properties:

AuditServers property	Gets the list of management databases in this installation.
CurrentAuditServer property	Gets the currently connected management database.
Name property (audit installation)	Gets the name of the audit installation.

Methods

The Installation class provides the following methods:

GetAuditStore method	Retrieves an audit store given its display name.
Publish method	Publishes installation information to Active Directory.

Discussion

An Installation object holds information about a specific audit installation. This class lets you retrieve information about an installation and publish changed information to the Active Directory domain controller so that it can be retrieved by the components of the installation.

See also

- GetInstallation method

AuditServers property

Gets the list of management databases in this installation.

Syntax

AuditServers class AuditServers {get;}

Return value

Returns the list of management databases in the installation.

Discussion

In most cases, an installation includes only one management database.

See also

- AuditServer class

CurrentAuditServer property

Gets the currently connected management database.

Syntax

AuditServer class CurrentAuditServer {get;}

Return value

Returns the connected management database.

Discussion

You can use the SQL Server instance name property of the management database object returned by this property as a parameter value when you call the Connection.GetInstallation (server,database) method.

See also

- [AuditServer class](https://docs.centrify.com/Content/aud-dbmgmt/AuditServerClass.htm)
- GetInstallation method

Name property (audit installation)

Gets the name of the audit installation.

Syntax

String Name {get;}

Return value

Returns the installation name.

Discussion

The audit installation is named when it is created and this name is not normally changed during the life of the installation.

GetAuditStore method

Retrieves an audit store given its display name.

Syntax

AuditStore class GetAuditStore(

string *Name*

)

Parameters

Errors

The GetAuditStore method may throw one of the following exceptions:

- Centrify.DirectAudit.Common.Logic.AuthenticationException if you do not have permission to connect to the Microsoft SQL Server instance or the

management database.

- Centrify.DirectAudit.Common.Logic.ConnectDatabaseException if you cannot connect to the Microsoft SQL Server instance either because the Microsoft SQL Server instance is not running and does not allow remote connections.

Example

The following code sample accepts the audit store display name as an argument when the script is executed, calls the GetAuditStore method to get the audit store, then attaches a new audit store database to the audit store:

```
...

strInstallationName = wscript.arguments.item(0)
strAuditStoreName = wscript.arguments.item(1)
strServerName = wscript.arguments.item(2)
strDatabaseName = wscript.arguments.item(3)

SET objConnection = CreateObject("Centrify.DirectAudit.Connection")
SET objInstallation = objConnection.GetInstallation(strInstallationName)
SET objAuditStore = objInstallation.GetAuditStore(strAuditStoreName)
today = Date
strDatabaseName = strDatabaseName & "-" & Year(today) & "-" & Month(today) &_
& "-" & Day(today)

SET objAuditStoreDatabase = objAuditStore.GetDatabase(strDatabaseName)

' Create a new Audit Store database and attach to the Audit Store
SET objAuditStoreDatabase = objAuditStore.AddDatabase(strDatabaseName, strServerName, strDatabaseName)
```

See also

- AuditStore class

Publish method

Publishes installation information to Active Directory.

Syntax

```
void Publish()
```

Errors

The Publish method may throw the following exception:

- Centrify.DirectAudit.Common.Logic.DirectAuditException if you do not have write permission for the installation's service connection point (SCP) object in Active Directory.

Discussion

Audit Manager publishes installation information to a service connection point (SCP) object in Active Directory so that audited computers and collectors can look up the information. For example, collectors publish which audit store they are part of so that once an agent determines which audit store is to receive its audit data, it can determine the list of collectors that service that audit store by querying Active Directory.

When you use the methods in the API to change settings in the installation, you must call the Publish method to write the new settings to the Active Directory domain controller so that other auditing components in the installation can find the new information.

Example

The following code sample illustrates using Installation.Publish in a script:

...

objInstallation.Publish
wscript.echo "Published settings to Active Directory."

Find Sessions is a separate executable file, installed in the same directory as Audit Analyzer, that you can use to find and open audited sessions. The program provides a graphical user interface and a command line interface for specifying the search criteria. You can use either interface to find sessions of interest. From the Find Sessions graphical user interface, you can also replay, update the review status, view the desktops used for any sessions found, display the list of indexed commands or events, and copy the session URI.

Starting Find Sessions

You can start Find Sessions from the Windows command line, using a web browser, or by selecting the View DirectAudit Sessions menu option in other applications, such as Access Manager and Active Directory Users and Computers.

For example, in Access Manager or Active Directory Users and Computers, you can select a computer or user, right-click, then select View DirectAudit Sessions to open Find Sessions. To start Find Sessions from the Windows command line, you can navigate to the Audit Analyzer installation directory and run the following command in a command prompt window:

```
findsessions /ia
```

Find Sessions Return Codes

For your reference, Find Sessions supports the following return codes to report the status of an operation performed:

0	The operation was successful.
1	The operation failed because Find Sessions could not parse the Session URI.
2	The operation failed because Find Sessions could not parse the user input.
3	The operation failed because Quick queries are not supported.
4	The operation failed because there were errors in the AQL format.
5	The operation failed because of an incompatible version of AQL was detected.
6	The operation failed because no installation was selected.
7	The operation failed because the installation specified was not found.
8	The operation failed because the AQL string contains the <group by> keyword.
9	The operation failed because no sessions were selected.
10	The operation failed because Find Sessions could not export the list of events.
11	The operation failed because Find Sessions could not export the session list.
12	The operation failed because Find Sessions could not export UNIX input or output.
13	Not all selected sessions were deleted.
14	An unknown error occurred.

Specifying the Sessions to Find

After you start Find Sessions by selecting View DirectAudit Sessions, from the Windows command line, or in a web browser, the program displays a graphical user interface for selecting search criteria. You can use the Common or Advanced search criteria to find sessions of interest. The Find Sessions dialog box then displays the results that match the criteria you specify. You can then replay, update the review status, display the list of indexed commands or events, copy session URI, or view the desktops used in any of the sessions returned.

In most cases, you can find the sessions you are interested in through some combination of user name, computer name, and session time displayed on the Common tab. If you right-click to View DirectAudit Sessions from a specific computer or user, that computer or user is automatically defined as the search criteria. If you want to specify additional criteria, such as review status or auditor name, you can click the Advanced tab.

To specify criteria by which to find sessions:

1. Start Find Sessions.
2. Select the desired installation from the Installation list.
3. On the Common tab, enter the basic search criteria as applicable for the sessions you want to find:
    1. User: Type all or part of the user name to find sessions for a particular user account.
    2. Machine: Type all or part of the computer name to find sessions run on a particular computer.
    3. Session start time: Select this option to find sessions based on when the session started. If you select this option, you can refine the search to include sessions started or not started in a specific number of days, hours, or minutes, or to include sessions started or not started today, yesterday, this week, last week, this month, last month, this year, or last year.
4. Click **Find Now** to find the sessions that match the criteria you specified.
5. Click **Clear All** to start a new query.

Specifying the Sessions to Find

After you start Find Sessions by selecting View DirectAudit Sessions, from the Windows command line, or in a web browser, the program displays a graphical user interface for selecting search criteria. You can use the Common or Advanced search criteria to find sessions of interest. The Find Sessions dialog box then displays the results that match the criteria you specify. You can then replay, update the review status, display the list of indexed commands or events, copy session URI, or view the desktops used in any of the sessions returned.

In most cases, you can find the sessions you are interested in through some combination of user name, computer name, and session time displayed on the Common tab. If you right-click to View DirectAudit Sessions from a specific computer or user, that computer or user is automatically defined as the search criteria. If you want to specify additional criteria, such as review status or auditor name, you can click the Advanced tab.

To Specify Criteria to Find Sessions:

1. Start **Find Sessions**.
2. Select the desired installation from the **Installation list**.
3. On the **Common** tab, enter the basic search criteria as applicable for the sessions you want to find:
    - User: Type all or part of the user name to find sessions for a particular user account.
    - Machine: Type all or part of the computer name to find sessions run on a particular computer.
    - Session start time: Select this option to find sessions based on when the session started. If you select this option, you can refine the search to include sessions started or not started in a specific number of days, hours, or minutes, or to include sessions started or not started today, yesterday, this week, last week, this month, last month, this year, or last year.
4. Click **Find Now** to find the sessions that match the criteria you specified.
5. Click **Clear All** to start a new query.

Specifying Advanced Criteria

In some cases, you might want to specify additional criteria for a search or to search exclusively on an attribute not found on the Common tab. For example, you might want to find only those sessions that have yet to be reviewed or all of the sessions where a specific command or application was used. To add criteria or perform these types of specialized searches, you can click the **Advanced** tab.

To Specify Advanced Criteria for Finding Sessions:

1. Start **Find Sessions**.
2. Select the desired installation from the **Installation list**.
3. Click the **Advanced** tab.
4. Click **Add** to add a new criterion.
5. Select an appropriate attribute from the Attribute list based on the sessions you want to find.

   For example: You can search for sessions based on the period of time in which they were active or based on a specific state. You can also search for sessions based on the activity that took place during the session. For example, you can find sessions where specific UNIX commands or Windows applications were used.
6. Select the appropriate criteria for the attribute you selected, then click **OK**.
7. The specific selections you can make depend on the attribute selected. For example, if the attribute is Review Status, you can choose Equals and the review state you want to find. If you select the attribute Comment, you can specify Contains any of and type the string that you want to find any part of.
8. When searching for user names or computers on the **Advanced** tab, use the Starts with option. If you use the default to match exactly, you must include the fully qualified domain name of the user or computer.
9. Click **Add** to add another criterion until you have defined all of the attributes for which you want to find sessions.
10. Click **Find Now** to find the sessions that match the criteria you specified.
11. Click **Clear All** to start a new query.

Adding Advanced Criteria

If you have more than one advanced criteria, different criteria attributes, such as Session Time and State, are separated by an implicit AND operation. Onl sessions that match both criteria are returned. If you have repeated criteria attributes, for example, time is not in past 10 days; time is in last month, the attributes are separated by an implicit OR operation. Sessions that match either criteria are returned.

Editing and Removing Advanced Criteria

You can edit and remove any of the advanced criteria you specify in Find Sessions. For example, if you are not finding the appropriate sessions, you might need to change or remove the criteria you have defined.

To Edit or Remove Find Sessions Criteria:

1. Start Find Sessions.
2. Select the desired installation from the Installation list.
3. Click the **Advanced** tab.
4. Select the criterion in the list of Define Criteria.
5. Click **Edit** to modify the definition or **Remove** to remove the criterion.

Finding Sessions From a Command Line

You can run Find Sessions as a command line utility on computers where Audit Analyzer is installed. The command line interface can be useful, for example, if you may want to find, export, or delete sessions as part of a script.

You can view usage information for the command line interface using the /help option.

To Use the Command Line Interface for Find Sessions:

1. Open a Command window and navigate to the Audit Analyzer directory.

   ```
   cd “C:\\Program Files\\Centrify\\Audit\\AuditAnalyzer”
   ```

2. Run the findsessions command with the /help option to view usage information.

   ```
   findsessions /help
   ```

3. Specify search criteria for finding sessions using the following format:

   ```
   findsessions /i=”*InstallationName*” /u=”*username*” /m=”*computerName*”/t=”*yyyy-MM-dd HH:mm:ss*”
   ```

   The installation name is required. You must also specify at least one of the other criteria (user name, computer name, or time). You can also combine the search criteria to refine your search.

   For user name and computer name, you can specify a portion of a name to find all sessions matching that name portion. For time, if you specify a date without a time, the assumed time is 12 midnight. For example, if you do the following search and you have sessions on computers named “KH-Win7” and “KH-W8,” the results include sessions for both computers.

   ```
   FindSessions /i=”DefaultInstallation” /m=”KH-W”
   ```

   The following example finds sessions for “Admin” and “Administrator” users:

   ```
   FindSessions /i=”DefaultInstallation” /u=”Admin”
   ```

   The following example finds sessions that were running at a specific time regardless of what time the sessions started or ended:

   ```
   FindSessions /i=”DefaultInstallation” /t=”2015-01-21 5:25:00”
   ```

   You can also find sessions for multiple users or computers by separating the user names or computer names using a semi-colon (;).For example, to search for audited sessions for the users maya and fred, you can specify both users in the command line like this:

   ```
   FindSessions /i=”DefaultInstallation” /u=”maya;fred”
   ```

   For more complex queries, you can also use AQL syntax on the command line.

   For details, see Finding sessions using AQL syntax.

Delinea

Find Sessions Command Line Usage Examples

You can view usage information for the command line interface using the /help option. That information is included here as well.

Usage:

FindSessions.exe [Connection] [Query] [Action] [Parameter]

Connection:
/i=<installation name> or /installation=<installation name>
Make a connection to the specified DirectAudit Installation.

Query:
Query can be defined by AQL or individual search criteria

/a=<aql statement> or /aql=<aql statement>
Use the specified AQL as a search criteria to find the audited sessions from DirectAudit databases.
This option should not be used together with /user, /machine or /activetime.

/u=<user name> or /user=<user name>
Find all audited sessions for a particular user from DirectAudit databases.
This option can be used together with /machine and /activetime, which means the returned sessions need to fulfill all specified criteria.
This option should not be used with /aql option.

/m=<machine name> or /machine=<machine name>
Find all audited sessions for a particular machine from DirectAudit databases.

This option can be used together with /user and /activetime, which means the returned sessions need to fulfill all specified criteria. This option should not be used with /aql option/t=<time> or /activetime=<time>.

/t=<time> or /activetime=<time>
Find all active audited sessions at a particular time from DirectAudit databases. This option can be used together with /user and /machine, which means the returned sessions need
to fulfill all specified criteria. This option should not be used with the /aql option.

/r="role1;role2" or /role="role1;role2"
Find all sessions with role role1 OR role2. Must be used with /export="UnixCommandlUnixInputlUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

/k="ticket1;ticket2" or /ticket="ticket1"
Find all sessions with trouble ticket ticket1 OR ticket2. Must be used with /export="UnixCommandlUnixInputlUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

Action:
/delete
Delete the sessions by the query.

/export=[SessionListlWashEventslUnixCommandlUnixInputlUnixInputOutput]
Export the sessions by the query. This option should used with /path option.

Parameter:
/path
Folder to save the export files. This option should used with /export option

/format=[htmllhtmlcsvlpdflxml]
Export the session list. this option should used with /export=SessionInfo /path=<folder path>

/suppresswarning
Suppress warning messages.

/onerror=[continue]
Continue processing session list if one or more databases are unreachable.

Examples:

FindSessions /installation="installation sample" /aql="1 time is in this week"

FindSessions /installation="installation sample" /aql="1 inputcommand = \"dzdo*\"\"" /delete

FindSessions /installation="installation sample" /aql="1 text = \"dzdo\"\"" /suppresswarning
/export="UnixInput" /path="folder path"

FindSessions /installation="installation sample" /user="user sample" /machine="machine sample"
/activetime="2011-12-24 15:30:45"

FindSessions /installation="installation sample" /aql="1 module = \"Windows PowerShell\"\""
/export="SessionList" /format="html" /path="folder path"

FindSessions /installation="installation sample" /aql="1 time is in this month"

/export="UnixInputOutput" /path="folder path" /role="role1;role2" /ticket="ticket1;ticket2"

Note: If the last field that you're search for includes double quotes, you need to escape the quotes. For example, findsessions -i="MyInstallation" /aql="1 time is in this week" doesn't have this issue but FindSessions /i="MyInstallation" /a="1 sessionid = \"a4006f206465-4db1-a2e7-a4e1f646c835\"\" does.

Editing and removing advanced criteria

You can edit and remove any of the advanced criteria you specify in Find Sessions. For example, if you are not finding the appropriate sessions, you might need to change or remove the criteria you have defined.

To edit or remove Find Sessions criteria:

1. Start Find Sessions.
2. Select the desired installation from the Installation list.
3. Click the **Advanced** tab.
4. Select the criterion in the list of Define Criteria.
5. Click **Edit** to modify the definition or **Remove** to remove the criterion.

Delinea

Finding sessions from a command line

You can run Find Sessions as a command line utility on computers where Audit Analyzer is installed. The command line interface can be useful, for example, if you may want to find, export, or delete sessions as part of a script.

You can view usage information for the command line interface using the /help option.

To use the command line interface for Find Sessions:

1. Open a Command window and navigate to the Audit Analyzer directory.
2. cd "C:\Program Files\Centrify\Audit\AuditAnalyzer"
3. Run the findsessions command with the /help option to view usage information.
4. findsessions /help
5. Specify search criteria for finding sessions using the following format:
6. findsessions /i="*InstallationName*" /u="*username*" /m="*computerName*" /t="*yyyy-MM-dd HH:mm:ss*"
7. The installation name is required. You must also specify at least one of the other criteria (user name, computer name, or time). You can also combine the search criteria to refine your search.
8. For user name and computer name, you can specify a portion of a name to find all sessions matching that name portion. For time, if you specify a date without a time, the assumed time is 12 midnight. For example, if you do the following search and you have sessions on computers named "KH-Win7" and "KH-W8," the results include sessions for both computers.
9. FindSessions /i="DefaultInstallation" /m="KH-W"
10. The following example finds sessions for "Admin" and "Administrator" users:
11. FindSessions /i="DefaultInstallation" /u="Admin"
12. The following example finds sessions that were running at a specific time regardless of what time the sessions started or ended:
13. FindSessions /i="DefaultInstallation" /t="2015-01-21 5:25:00"
14. You can also find sessions for multiple users or computers by separating the user names or computer names using a semi-colon (;).For example, to search for audited sessions for the users maya and fred, you can specify both users in the command line like this:
15. FindSessions /i="DefaultInstallation" /u="maya;fred"

For more complex queries, you can also use AQL syntax on the command line. For details, see Finding sessions using AQL syntax.

Find sessions command line usage examples

You can view usage information for the command line interface using the /help option. That information is included here as well.

Usage:

FindSessions.exe [Connection] [Query] [Action] [Parameter]

Connection: /i=<installation name> or /installation=<installation name>

Make a connection to the specified DirectAudit Installation.

Query:
Query can be defined by AQL or individual search criteria

/a=<aql statement> or /aql=<aql statement>
Use the specified AQL as a search criteria to find the audited sessions from DirectAudit databases.
This option should not be used together with /user, /machine or /activetime.

/u=<user name> or /user=<user name>
Find all audited sessions for a particular user from DirectAudit databases.
This option can be used together with /machine and /activetime, which means the returned sessions need to fulfill all specified criteria.
This option should not be used with /aql option.

/m=<machine name> or /machine=<machine name>
Find all audited sessions for a particular machine from DirectAudit databases.

This option can be used together with /user and /activetime, which means the returned sessions need to fulfill all specified criteria. This option should not be used with /aql option/t=<time> or /activetime=<time>.

/t=<time> or /activetime=<time>
Find all active audited sessions at a particular time from DirectAudit databases. This option can be used together with /user and /machine, which means the returned sessions need
to fulfill all specified criteria. This option should not be used with the /aql option.

/r="role1;role2" or /role="role1;role2"
Find all sessions with role role1 OR role2. Must be used with /export="UnixCommandlUnixInputlUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

/k="ticket1;ticket2" or /ticket="ticket1"
Find all sessions with trouble ticket ticket1 OR ticket2. Must be used with /export="UnixCommandlUnixInputlUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

Action:
/delete
Delete the sessions by the query.

/export=[SessionListlWashEventslUnixCommandlUnixInputlUnixInputOutput]
Export the sessions by the query. This option should used with /path option.

Parameter:
/path
Folder to save the export files. This option should used with /export option

/format=[htmllhtmlcsvlpdflxml]
Export the session list. this option should used with /export=Sessionlnfo /path=<folder path>

/suppresswarning
Suppress warning messages.

/onerror=[continue]
Continue processing session list if one or more databases are unreachable.

Examples:

FindSessions /installation="installation sample" /aql="1 time is in this week"

FindSessions /installation="installation sample" /aql="1 inputcommand = \"dzdo*\"\"" /delete

FindSessions /installation="installation sample" /aql="1 text = \"dzdo\"\"" /suppresswarning
/export="UnixInput" /path="folder path"

FindSessions /installation="installation sample" /user="user sample" /machine="machine sample"
/activetime="2011-12-24 15:30:45"

FindSessions /installation="installation sample" /aql="1 module = \"Windows PowerShell\"\""
/export="SessionList" /format="html" /path="folder path"

FindSessions /installation="installation sample" /aql="1 time is in this month"

/export="UnixInputOutput" /path="folder path" /role="role1;role2" /ticket="ticket1;ticket2"

Note:If the last field that you're search for includes double quotes,
you need to escape the quotes. For example, findsessions -i="MyInstallation"
/aql="1 time is in this week" doesn't have this issue but FindSessions
/i="MyInstallation" /a="1 sessionid =
\\"a4006f206465-4db1-a2e7-a4e1f646c835\\"\\" does.

Finding sessions using AQL syntax

If you are an experienced programmer and want to write complex queries, you can use AQL statements on the command line.

To use AQL to find sessions at the command line:

1. Open a command window and navigate to the Audit Analyzer directory.

   cd "C:\Program Files\Centrify\Audit\AuditAnalyzer"

2. Run the findsessions command with the following syntax:

   FindSessions /i="*InstallationName*" /aql="*AQL query text*"

   For example, the following is a simple query that searches for sessions that were running in the current week:

   findsessions -i="MyInstallation" /aql="1 time is in this week"

   To find a specific session using the session identifier, you might write a query similar to the following:

   FindSessions /i="MyInstallation" /a="1 sessionid = \"a4006f206465-4db1-a2e7-a4e1f646c835\"\"

   To find a specific session using the user display name, you might write a query similar to the following:

   findsessions /i="installationname" /a="1 displayname=\"maya*\"\"

Note: When you enter a search term, AQL looks for an exact match. To search for sessions that start with the term you entered, add an asterisk to the search term. For example, user=\\"maya*\\"\\" finds sessions for users such as maya@acme.com, mayan@acme.com, and so forth. Otherwise, a search for user=\\"maya\\"\\" returns nothing and a search for user=\\"maya@acme.com\\"\\" returns sessions for just that one user.

Note: If the last field that you're search for includes double quotes, you need to escape the quotes. For example, findsessions \-i="MyInstallation" /aql="1 time is in this week" doesn't have this issue but FindSessions /i="MyInstallation" /a="1 sessionid = \\"a4006f206465-4db1-a2e7-a4e1f646c835\\"\\" does.

Simplifying AQL queries

Writing valid AQL queries for the command line can be challenging. The basic format for AQL statements in Backus-Naur notation consists of the following parts:

<aql> ::= <version> {<quick_terms>} | {<type> | <filter>}

To simplify the process of generating the AQL queries you want to use on the command line, you can use Audit Analyzer to create a new private query and use the user interface to specify the query criteria. After you have created the query, you can right-click the query node, and click **Export Query Definition** to save the query definition as a file. You can then extract the AQL statement from the query definition. You can then delete the private query node from Audit Analyzer if it is not needed.

For example, run the command with the definition from the private query:

findsessions -i="MyInstallation" /aql="1 type= shellui, wingui; time is in this week; review = Reviewed"

Delinea

Audit Query Language overview

You can use the Audit Query Language to search for audited sessions with Find Sessions from a command line interface.

The Audit Query Language (AQL) serves two purposes:

- **Query definition:** The Audit Management Server database stores the query definition as an AQL statement.
- **Query language:** In order to query for audit information, the audit & monitoring service sends AQL statements to the Audit Management Server database.

When you enter an AQL query, the system stores this as the query definition. The query definition defines what information is of interest and how to group the results. In some cases, you might retrieve the results over multiple phases, depending on how you want to present the information.

For example: The query "get all Windows audit sessions, grouped by user" has two phases:

1. Gather a list of all users who have Windows audit sessions
2. Show all the Windows sessions for each user who is listed in Step 1.

In each phase, the Audit & Monitoring generates the AQL statement and sends it to the Audit Management Server database in order to query for audit information. This part is when the AQL statements function as a query language.

Here is an AQL statement example:

1 Type=wingui; orderby=time DESC; time is in this week; user="joe*","mark*";machine="domaincontroller"'

The example query would return audited Windows sessions in the last week where Joe or Mark logged in to the domain controller system, and the results would be listed in descending order of when they occurred.

In general, the format of an AQL statement can either be just some quick search terms or a statement with the following parts:

- Audit trail types
- Group-by
- Order-by
- Predicates

Backus-Naur Form (BNF) definition of AQL

Here is the Backus-Naur Form (BNF) definition of the AQL language syntax so that you can see how the query language is constructed.

<aql> ::= <version> {<quick_terms>} | {<type> | <groupby> | <filter>}

<version> ::= any numeric number. Currently, we support only 1.

<quick_terms> ::= <word> | <and_words> | <or_words> | <exact_combined_words>

<and_words> ::= <word> (" " <word>)+

<or_words> ::= <word> ("OR" <word>)+

<exact_combined_words> ::= """ <and_words> """

<word> ::= any printable string except white spaces

<type> ::= "type" ("=" | ":") <typename> {"," <typename>}

<groupby> ::= "groupby" ("=" | ":") <groupname> {"," <groupname>}

<orderby> ::= “orderby” (“=”|”:”) <columnname> {“ASC”|”DESC”}

<filter> ::= <normal_filter> | <negative_filter>

<normal_filter> ::= <string_filter> | <time_filter> | <number_filter> | <enum_filter> |

<boolean_filter> | <ip_filter>

<negative_filter> :: = "not(" <normal_filter> ")"

<string_filter> :: = <string_field> <string_op> """ <string_val> """ {"," """ <string_val> """}

<string_op> ::= "=" | "!="

<time_filter> ::= <single_time_filter> | <between_time_filter> | <in_predefined_time_filter> |

<in_past_filter>

<single_time_filter> ::= <time_field> <single_time_op> <single_time_val> <!-- single_time_val is in

format of "yyyy-mm-dd hh:mm:ss" -->

<time_op> :: = "is before" | "is after" | "is not before" | "is not after"

<between_time_filter> ::= <time_field> <between_time_op> <between_time_val>

<between_time_op> ::= "is between" | "is not between"

<between_time_val> ::= <single_time_val> " and " <single_time_val>

<in_predefined_time_filter> ::= <time_field> <in_predefined_time_op> <predefined_time_val>

<in_predefined_time_op> ::= "is in" | "is not in"

<predefined_time_val> ::= "today" | "yesterday" | "this week" | "last week" | "this month" | "last month" |

"this year" | "last year"

<in_past_filter> ::= <time_field> <in_past_op> <digit>+ <unit_of_time>

<in_past_op> ::= "is in_past" | "is not in_past"

<unit_of_time> ::= "day" | "hour" | "minute"

Note:Currently, AQL has filters only for strings and time.

AQL usage examples

AQL usage examples

/i

/installation

/aql

/a

/user

/u

/machine

/m

/activetime/

/t

/suppresswarning

/sw

/delete

/export

/r

/role

/path

/format

/ticket

/k

Find Sessions Usage formats:

FindSessions /i=<installationName> /a=<AQL query>

FindSessions /i=<installationName> /u=<user or semi-colon-separated list of users> /m=<machine or semi-colon-separated list of machines> /t=<YYYY-MM-DD HH:MM:SS>

FindSessions /i=<installationName> /a=<AQL query> /

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList> /format=<html|htm|csv|pdf|xml> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList|WashEvents|UnixCommand|UnixInput|UnixInputOutput> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<UnixCommand|UnixInput\UnixInputOutput> /sw /path=<folderpath> /r="role1;role2" /"ticket1;ticket2"

Find Sessions: Usage examples without AQL:

FindSessions /i="DirectAudit" /user="user sample" /machine="machine sample" /activetime="2018-12-24 15:30:45"

FindSessions /i="DirectAudit" /user="maya;fred" /machine="KH-Win7;KH-Win8" /activetime="2018-12-24 15:30:45"

Find Sessions: Usage examples with AQL:

FindSessions /i="DirectAudit" /aql="1 time is in this week"

FindSessions /i="DirectAudit" /aql="1 module = \"Windows PowerShell\"\"" /delete

FindSessions /i="DirectAudit" /aql="1 text=\"dzdo\"\"" /export="UnixCommand" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 inputcommand = \"dzdo*\"\"" /suppresswarning /export="UnixInputOutput" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 sessionid=\"D108F7B2-F4FB-FB42-A6E7-A40454780690\"\"

AQL quick search terms

ou can just enter a series of keywords if you just want to do a quick search.

Here are some examples:

- joe : any data fields that contain the word 'joe'
- joe john : any data fields that contain both words 'joe' and 'john'
- joe OR john : any data fields that contain 'joe' or 'john'
- "joe john" : any data fields that contain the exact phrase “joe john”

The default operator designated by a space between terms is evaluated as an “AND” operator, so there is no need to include “AND” between terms. Explicit operator takes precedence over implicit operator, thus “OR” is always evaluated before the absence of an operator.

Here are some examples of quick search term queries in AQL:

The database searches the following data fields with the keywords in quick search terms:

- User (username)
- Machine (machine name)
- Time (audit trail data record start time)
- Module
- Text

AQL audit trail types example

AQL audit trail types example

If desired, you can refer to the types of audit trails to include in the results, such as Windows sessions or UNIX sessions. You can specify one or more audit trail types with the "type:" parameter.

If you don't include this parameter, the results include all audit trail types.

If you specify more than audit trail type, the results are those that fit all specified parameters.

Example:

type=wingui, shellui

AQL group-by example

If desired, you can specify how to group the results.

Example:

groupby=user, date

When you specify multiple groupby criteria, the database groups the results by the first criterion and displays the immediate result by ignoring the remaining criteria. When you double-click the results, then the database displays more results according to the remaining criteria.

Note:FindSessions does not support the use of groupby.

AQL order-by example

If desired, you can specify how to sort the AQL query results by using orderby.

For example:

orderby=time, user ASC, machine DESC

The sort order options are as follows:

- ASC: sort results in ascending order
- DESC: sort results in descending order

If you don't specify a sort order, the system uses 'ASC' by default.

AQL Predicates

Using predicates in your AQL query is entirely optional. You can filter the result set by any number of predicates or none at all. Each predicate expresses a condition that must be true in order for the service to include a record in the result set.

There is an implicit 'AND' between each predicate. If you repeat a predicate for a field, there is an implicit 'OR' between them.

Each predicate refers to a field in the schema of an audit trail type. If the field name does not specify an audit trail type, then the field must exist for all selected audit trail types. If the field name specifies an audit trail type specified with the "type:" parameter, then the predicate applies only to that audit trail type.

AQL predicate behavior examples

Example A: Type=wingui, shellui; user = “joe”

The above example selects all Windows and UNIX sessions for joe.

Example B: Type=wingui, shellui; shellui.user = “joe”

The above example selects all Windows sessions but only UNIX sessions for joe.

The service categorizes predicates according to the field data type:

- String
- Number
- Boolean
- Date / time
- IP
- Enumeration

AQL string predicate behavior

Here are some examples of how to filter an AQL query based on string predicates:

field = "<string>","<string2>",...	exact match
field != "<string>", “<string2>”,...	not equals (exact match)
field = "<string>*", “<string2>*”,...	starts with
field != "<string>*", “<string2>*”,...	not starts with

AQL number predicate behavior

Here are some examples of how to filter an AQL query based on number predicates:

field = <number>	equals
field != <number>	not equal
field >= <number>	greater than or equal
field > <number>	greater than
field <= <number>	smaller than or equal

field < <number>	smaller than

You can replace <number> with any integer or floating point number, such as 1 or -3.14.

AQL boolean predicate behavior

Here are some examples of how to filter an AQL query based on boolean predicates:

field = true	true
field != false>	false

AQL Date and time predicate behavior

Here are some examples of how to filter an AQL query based on date and time predicates:

field is (not) before <datetime>	before a specific date and time or not
field is (not) after <datetime>	after a specific date and time or not
field is (not) between <datetime> <datetime>	between two dates and time or not
field is (not) in_past <number> <unit>	in the past period of time or not, where the unit is day, hour, or minute
field is (not) in <predefined time>	field is not in the predefined time or not

Replace <datetime> with a particular date and time with the following format:

- Y-M-D, for example 2019-12-15
- Y-M-D h:m:s, for example 2019-12-15 15:30:00

Replace <predefined time> with any of the following values:

- today
- yesterday
- this week
- last week
- this month
- last month
- this year
- last year

AQL IP predicate behavior

Here are some examples of how to filter an AQL query based on IP address predicates:

field = <ip>	equals
field != <ip>	not equal
field >= <ip>	greater than or equal
field > <ip>	greater than
field <= <ip>	smaller than or equal
field < <ip>	smaller than

AQL enumeration predicate behavior

Here are some examples of how to filter an AQL query based on enum predicates:

field = <enum>	equals
field != <enum>	not equal

Replace <enum> with the values appropriate for the field you're querying against.

For example, filtering for a session state involves specifying an enum value:

state = Terminated

state != InProgress

AQL keywords

Session time	Date/Time predicate
UNIX command time	Date/Time predicate
State	Enum predicate: Unknown, InProgress, Terminated, Disconnected, Completed, ToBeDeleted
Review status	Enum predicate: None, ToBeReviewed, Reviewed, PendingForAction, KeepForever, ToBeDeleted
Session size	Numeric predicate (in kilobytes)
Unix outputs and commands	String predicate
User	String predicate
Machine	String predicate
Auditstore	Number predicate
Parameters of commands and applications	String predicate
Unix command name	String predicate

Windows applications	String predicate
Comment	String predicate
Session Id	String predicate
Client name	String predicate
User display name	String predicate
Account	String predicate
Text	String predicate
Module	String predicate
Tag	String predicate

AQL usage examples

/i

/installation

/aql

/a

/user

/u

/machine

/m

/activetime/

/t

/suppresswarning

/sw

/delete

/export

/r

/role

/path

/format

/ticket

/k

Find Sessions Usage formats:

FindSessions /i=<installationName> /a=<AQL query>

FindSessions /i=<installationName> /u=<user or semi-colon-separated list of users> /m=<machine or semi-colon-separated list of machines> /t=<YYYY-MM-DD HH:MM:SS>

FindSessions /i=<installationName> /a=<AQL query> /

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList> /format=<html|htm|csv|pdf|xml> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList|WashEvents|UnixCommand|UnixInput|UnixInputOutput> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<UnixCommand|UnixInput\UnixInputOutput> /sw /path=<folderpath> /r="role1;role2" /"ticket1;ticket2"

Find Sessions: Usage examples without AQL:

FindSessions /i="DirectAudit" /user="user sample" /machine="machine sample" /activetime="2018-12-24 15:30:45"

FindSessions /i="DirectAudit" /user="maya;fred" /machine="KH-Win7;KH-Win8" /activetime="2018-12-24 15:30:45"

Find Sessions: Usage examples with AQL:

FindSessions /i="DirectAudit" /aql="1 time is in this week"

FindSessions /i="DirectAudit" /aql="1 module = \"Windows PowerShell\"\"" /delete

FindSessions /i="DirectAudit" /aql="1 text=\"dzdo\"\"" /export="UnixCommand" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 inputcommand = \"dzdo*\"\"" /suppresswarning /export="UnixInputOutput" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 sessionid=\"D108F7B2-F4FB-FB42-A6E7-A40454780690\"\"

Accessing sessions via web browser

On computers that have Audit Analyzer installed, you can also find and play back sessions from a web browser. Because the cda:// protocol is automatically registered on the computer with Audit Analyzer, you can use a web browser to open Find Sessions or to replay a specific session. For example, you can embed a cda:// link in a web page to automatically generate a list of sessions, or you might want to embed a link to a session or set of sessions in a web-based report or event notification.

Opening Find Sessions from a web browser

You must be able to specify a query using AQL syntax to open Find Sessions from a web browser. If you want to start playing back a session from a web browser, you must know the session identifier. You can extract the session identifier from the session URI.

To start Find Sessions from a web browser:

1. Open a web browser.
2. Type the installation name and a search string using AQL syntax in the address bar of the web browser.
3. For example, if you want to search an installation named MyInstallation5 for sessions that involved the Administrator user, you would type the following in the address bar:
4. cda://MyInstallation5/?search=\"1 user=\"Administrator*\"\
5. Click **Allow** to open the Find Sessions with the Advanced tab displayed and "user=Administrator*" listed for the Define Criteria.
6. Click **Find Now** to find sessions matching the criteria you specified.

Playing back a session from a web browser

If you want to start playing back a session from a web browser, you must know the session identifier. You can extract the session identifier from the session URI.

To get the session identifier:

1. In the session player, select File > Copy Session URI.
2. Open a text editor and paste the session URI into the file.
3. Delete the portion of the URI that identifies the player and installation, so that only the object GUID remains.
4. For example, if the URI looks like this:
5. rep://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395
6. Remove the part of the URI so that you only have the session identifier:
7. b62bc280-678c-439a-aec3-09a9b7ee4395

To play back a specific session from a web browser:

1. Open a web browser.
2. Type the installation name and session ID in the address bar of the web browser:
3. cda://‹*installationName*›/‹*session_id*›
4. For example:
5. cda://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395
6. The session player opens and plays the specified session.

Exporting sessions and session data

In addition to specifying the criteria for finding sessions of interest, you can use Find Session to selectively export session data to a file. You can export the following information:

- A list of sessions matching the criteria you specify.
- An indexed list of events associated with the Windows sessions that match the criteria you specify.
- An indexed list of commands associated with the UNIX sessions that match the criteria you specify.
- The UNIX input associated with the UNIX sessions that match the criteria you specify.
- The UNIX input and output associated with the UNIX sessions that match the criteria you specify.

You specify the export operation, type of data to export, file format, and file location using the following command line options:

/export=[SessionList|WashEvents|UnixCommand|UnixInput
|UnixInputOutput]
/format=[html|htm|csv|pdf|xml]
/path=<folder_path>

You can use these options in combination with other criteria, such as /user or /machine, to export information for a specific user, computer, or time. You can specify the /format option used for exporting the sessions of interest. If you don't specify the /format option, sessions matching the criteria you specify are exported as comma-separated values (.csv) in a text file. If you are exporting Windows events, UNIX commands, UNIX input, or UNIX input and output, each session is exported as a separate file in the format you specify.

If you are exporting UNIX commands, UNIX input, or UNIX input and output, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. Before you can use these options, however, you must configure the information required. For example, if you want to find all of the UNIX commands executed by a user running the db_backup role, you must first define and assign the db_backup role using Access Manager.

Exporting a session list

To export a list of sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="SessionList"
/format="*format*" /path="*folder*"

For example, to export the session list for all users in HTML format and save the output in the C:\Temp\Exported Sessions folder, you would type a command like this:

FindSessions /i="MyInstallation" /export="SessionList"
/format="html" /path="C:\Temp\Exported Sessions"

The command generates the list of sessions in the format specified. In this case, the command would generate an HTML file named SessionList in the C:\Temp\Exported Sessions folder with the following information for each session exported:

- User name, display name, account used, computer name, and audit store for the session.
- Start time, end time, and current state of the session.
- Client name associated with the session.
- Review status, user who last modified the review status, the time the status was last modified, and the comment added when the session was last modified.
- Size of the session in KB.
- Session URI that can be used to replay the session.

Exporting Windows events

To export an indexed event list for Window sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="WashEvents" /path="*folder*"

For example, to export the indexed event list for the sessions associated with a specific user and save the output in the C:\Temp\Session Events folder, you would type a command like this:

FindSessions /i="MyInstallation" /user="chris.howard"
/export="WashEvents" /path="C:\Temp\Session Events"

The command generates the list of events as comma-separated values in a text file. For example:

"Time","Application","Title","Type","Desktop","Audited","Role","Ticket"

"1/29/2015 1:53:14 PM", "Windows Explorer", "Start", "Application Activate", "Default", "Y","<None>","<None>"
"1/29/2015 1:53:56 PM", "DirectAuthorize System Tray", "Options", "Application Activate", "Default", "Y","<None>","<None>"
...
"1/29/2015 3:00:51 PM", "Windows Explorer", "Start", "Window Activate", "LocalSQLAdmin", "Y","<None>","<None>"
"1/29/2015 3:01:16 PM", "Microsoft SQL Server Management Studio Express", "Microsoft SQL Server Management Studio Express", "Application Activate", "LocalSQLAdmin", "Y","<None>","<None>"
.

Exporting UNIX command lists

To export an indexed command list for UNIX sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixCommand" /path="*folder*"

For example, to export the indexed command for the sessions associated with a specific computer and save the output in the C:\Temp\UNIX folder, you would type a command like this:

FindSessions /i="MyInstallation" /machine="rhes-63"
/export="UnixCommand" /path="C:\Temp\UNIX"

The command generates the list of commands as comma-separated values in a text file. For example:

"Time","Command","Role","Ticket"
"10/9/2014 3:12:14 PM","/bin/bash ","<None>","<None>"
"10/9/2014 3:12:19 PM","adflush","<None>","<None>"
"10/9/2014 3:12:23 PM","su -","<None>","<None>"
"10/9/2014 3:12:27 PM","Password: ","<None>","<None>"
"10/9/2014 3:12:30 PM","adflush","<None>","<None>"
"10/9/2014 4:26:14 PM", "exit","<None>","<None>"

Searching for sessions by role or trouble-ticket information

When you use the /export=UnixCommand option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information.

Use /role to specify search criteria based on one or more privilege elevation service roles. You can specify multiple roles separated by semicolons (;). For example, add /role="db_backup/zonename;mail_admin/zonename" to the command line to search for UNIX sessions that were run using the db_backup or mail_admin role.

Tip: When you search for sessions by role name, be sure to include the zone name. Otherwise, FindSessions doesn't return the sessions and instead displays the message, "No session is selected to be exported".

FindSessions /i="MyInstallation" /export="UnixCommand"
/role="db_backup/zonename;mail_admin/zonename" /path="C:\Temp\UNIX"

You can use the /ticket option to specify search criteria based on the trouble-ticket information if you have configured in the dzcheck script to collect this information. You can specify multiple tickets separated by semicolons (;). For example, add /ticket="ticket 1;ticket 2" to the command line to search for sessions ticket1 or ticket2 were specified.

You cannot use wildcards to search for role names or ticket information. If you specify both the /role and /ticket options, FindSessions returns the sessions that match both the specified roles and the specified trouble-ticket information. For information about configuring the dzcheck script and how to capture trouble-ticket information, see the *Administrator's Guide for Linux and UNIX*.

Exporting UNIX input

To export UNIX input from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixInput" /path="*folder*"

For example, to export the UNIX input for a specific user and save the output in the C:\Temp\Input folder, you would type a command like this:

FindSessions /i="MyInstallation" /user="tai-u1" /export="UnixInput" /path="C:\Temp\Input"

The command exports UNIX input to a text file. For example:

"UnixInputData","Role","Ticket"
"[1/20/2015 4:13:38 PM] K: PS1=NetShell:<CR>","<None>","<None>"
"[1/20/2015 4:13:38 PM] K: stty kill ^u erase ^h<CR>","<None>","<None>"

"[1/20/2015 4:13:38 PM] K: TERM=dumb<CR>","<None>","<None>"
"[1/20/2015 4:13:38 PM] K: set TERM=dumb<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: cat /etc/passwd<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: echo $?<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: cat /etc/group<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: echo $?<CR>","<None>","<None>"

When you use the /export=UnixInput option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. For details about using these options, see Using Find Session.

Exporting UNIX input and output

To export UNNIX input and output from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixInputOutput" /path="*folder*"

For example, to export UNIX input and output for a specific computer and save the output in the C:\Temp\Output folder, you would type a command like this:

FindSessions /i="MyInstallation" /m="firefly-sf" /export="UnixInputOutput" /path="C:\Temp\Output"

The command exports UNIX input and output to a text file. For example:

"UnixInputOutputData","Role","Ticket"
"[1/21/2015 10:53:20 AM] 0: /bin/bash ","<None>","<None>"
"[1/21/2015 10:53:23 AM] 1: [maya@firefly-sf Desktop]$ pwd","<None>","<None>"
"[1/21/2015 10:53:23 AM] K: pwd<CR>","<None>","<None>"
"[1/21/2015 10:53:23 AM] 2: /home/maya/Desktop","<None>","<None>"
"[1/21/2015 10:53:34 AM] 3: [maya@firefly-sf Desktop]$ cd /tmp","<None>","<None>"
"[1/21/2015 10:53:34 AM] K: cd /tmp<CR>","<None>","<None>"
"[1/21/2015 10:53:54 AM] K: ls -al in*<CR>","<None>","<None>"
"[1/21/2015 10:53:54 AM] 4: [maya@firefly-sf tmp]$ ls -al in*","<None>","<None>"
"[1/21/2015 10:53:54 AM] 5: -r-xr-xr--. 1 root root 313027 Dec 16 05:51 install.sh", "<None>","<None>"
"[1/21/2015 10:54:04 AM] K: su -<CR>","<None>","<None>"
"[1/21/2015 10:54:04 AM] 6: [maya@firefly-sf tmp]$ su -","<None>","<None>"

"[1/21/2015 10:54:10 AM] 7: Password: ","<None>","<None>"
"[1/21/2015 10:54:10 AM] K: xxxxxxxx<CR>","<None>","<None>"

When you use the /export=UnixInputOutput option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. For details about using these options, see Using Find Sessions.

Suppressing warning messages

By default, Find Sessions will generate warning messages if you attempt to export sessions without expected activity. For example, if you run a command to export UNIX input and output using /export="UnixInputOutput" and there is no user input activity, you might see warning messages similar to the following:

Finished exporting the sessions successfully.

Warning, URI:rep://BLD08/f435d61c-f191-4344-8adf-9d1432cb35ea,
Message: There is no user inputs captured in this session.

You can safely suppress these warning messages using the /suppresswarning or /sw command line option. For example, you might run a command similar to this:

C:\AuditAnalyzer› findsessions /i="BLD08" /role="verify"
/format=csv /path="C:\Temp" /export="UnixInputOutput"
/a="1 time is in today" /suppresswarning

This command would export the UNIX output without displaying warning messages about there being no user input.

Deleting sessions

You can also use Find Sessions to delete sessions matching the criteria you specify from the command line. You can use the /delete option in combination with other criteria, such as /user or /machine, to delete information for a specific user, computer, or time. However, if you specify the /delete on the command line, all of the sessions returned by the query are deleted.

To delete sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /delete

For example, to delete the sessions for a specific user on a specific computer, you would type a command like this:

FindSessions /i="MyInstallation" /user="tai-u1"
/machine="rhes63" /delete

Note that you cannot use the /delete option in combination with the /export option. If you want to export session information before deleting, you must do so in two separate operations.

Sample script for deleting multiple sessions

You can use Find Sessions to delete multiple sessions manually from the command line or using Windows Task Scheduler to automate the task. However, if you are deleting multiple sessions at once, you might want to execute the command from a batch file to ensure that Find Sessions will wait for the operation to complete and return the result of the operation.

The following is a sample script to delete sessions from TestInstallation recorded in the current month.

```
----------Start of FindSessions_Delete.bat----------
@ECHO OFF
cd "C:\Program Files\Centrify\Audit\AuditAnalyzer"
Start /WAIT FindSessions.exe /i="TestInstallation" /a="1 time is in this month" /delete
if ERRORLEVEL 1 (goto FindSessionError)
goto Succeeded
:FindSessionError
echo
##################################################################################

echo ## FindSession execution failed. ErrorLevel: %ERRORLEVEL% ##
echo
##################################################################################

goto exit
:Succeeded
echo FindSession execution succeeded.
:exit
----------End of FindSessions_Delete.bat----------
```

You can use a similar batch file if you want to export multiple sessions at the same time. To write a script for exporting information, you would specify the type of information to export and the path for saving the exported output. For example, if you want to export UNIX commands for MyInstallation to the C:\UNIX folder, the script could include a command like this:

```
Start /WAIT FindSessions.exe /i="MyInstallation"
/export="UnixCommand" /path="C:\UNIX"
```

Using Windows and Linux/Unix

The following OS User Reference topics are available:

- User's Guide Linux/Unix
- User's Guide Windows

The following topics are covered:

Getting started

This chapter describes how to use Centrify to access applications and run commands with privileges on a UNIX or Linux computer that has the Centrify Agent installed.

Verify You can Log in

If an administrator has installed the Centrify Agent on a UNIX or Linux computer you use, the next step is to verify that you can log in successfully. The Centrify Agent does not change how you log in to your computer. However, you must be assigned at least one role that allows you to log in.

When you are prompted for a user name and password, type your Active Directory or UNIX user name and your Active Directory password. If you provide valid credentials and have been assigned a role with permission to log in, you should be able to log in to your computer with a standard UNIX shell. If this is a computer you used earlier, before it became a Centrify-managed computer, there should be no noticeable changes to your working environment.

As a part of the deployment of Centrify software, your computer may or may not have been joined to a zone. To verify that the Centrify Agent is installed, that you are connected to an Active Directory Domain, and that you are connected to a zone, run the adinfo command. For example, if you are a user named billy in a zone named KHeadquarters, your output may look similar to the following:

```
[billy@kh-rh Desktop]$ adinfo
Local host name: kh-rh
Joined to domain: demo.acme.com
Joined as: kh-rh.demo.acme.com
Pre-win2K name: kh-rh
Current DC: deploy.acme.com
Preferred site: Default-First-Site-Name
Zone: demo.acme.com/Program Data/Acme/Zones/KHeadquarters
CentrifyDC mode: connected
Licensed Features: Enabled
```

To learn more about commonly used commands that may be available to you, see Commands available for users.

If the Centrify Agent is installed but not connected to a zone, or if the agent is not installed on your local computer, you should contact your administrator.

If the zone information for the agent is configured, but the agent status is Disconnected, restart the agent.

To restart the agent type the following:

```
$ adclient -x
$ adclient
```

If the agent status is still Disconnected, contact your system administrator.

Multi-Factor Authentication

Your organization may require multi-factor authentication in order for you to log in to your computer, or to execute commands using elevated privileges (dzdo) in a normal or restricted shell (dzsh) environment.

If multi-factor authentication is required as part of the login process, you will have to provide a password as well as a second form of authentication to log in to your computer. If multi-factor authentication is required as part of a re-authentication process, such as when you use command rights with elevated privileges or in a restricted shell, you must provide a password and either one or two other forms of authentication other than a password.

Checking Your Rights and Role Assignments

Your role assignments control where you can log in, the type of account you use to log in, the specific access rights you have on local or network computers, the types of commands you can execute, and whether you must log in using a restricted shell. As discussed in Types of access rights, there are three categories of access rights for UNIX and Linux computers:

- Command rights
- PAM application rights
- Secure shell session-based rights

Depending on the details of how roles are defined in your organization and the specific roles you have been assigned, you might have some or all of the access rights described in the following sections.

You can use the dzinfo command to look up detailed information about your rights and role assignments, any restrictions on when they are available, and what the roles allow you to do. To learn more about the dzinfo command, see Check your rights and roles using dzinfo.

> **Note**: You can view information about your own access rights and role assignments only.

Working with Command Rights

Command rights allow you to use commands to perform specific operations. The most basic rights—such as the right to log in—are defined when your administrator defines roles. Other, more granular command rights control access to individual command-line programs.

Using Command Rights in a Standard Shell

Command rights are assigned to you so that you can perform privileged operations that are not available to you by default.

On most UNIX and Linux computers, commands that require elevated permissions can be run by invoking the sudo command. The Centrify Agent provides similar functionality, but the commands are instead invoked using the dzdo command, then typing the command to execute, including any command-line options that you are allowed to use.

For example, assume your administrator has defined a command right for adjoin that enables you to execute the command as the root user. If this right is added to a role that has been assigned to you, you can execute the command by typing the following:

dzdo adjoin

Using Command Rights in a Restricted Shell Environment

Centrify provides a customized Bourne shell, dzsh, to serve as a restricted shell environment that is used to limit what commands you can execute for certain roles. For most operations, working in the dzsh shell is similar to working in an unrestricted shell except that the command set is limited to the command rights added by the administrator.

After your administrator has defined command rights, added them to role definitions, and assigned the roles to you, you can execute those commands in a restricted shell environment by typing the command, including any command-line options you are allowed to use. When you are finished running the command, you can switch back to your standard shell if you have the appropriate login right on that computer.

For example, assume that on your own computer, you can run the adinfo command in the standard shell, but you need to execute the command on a computer that is not yours. Your administrator has assigned you a role, AdminADinfo that grants you a UNIX login right and a right that requires you to run the adinfo command in a restricted shell on the computer you need to access. You must switch to this role to run the command on the specified computer. To do this, you log in to the computer you want to access and select the role your administrator has assigned to you. If you are a member of the zone Headquarters, you would type the following:

```
$ dzsh
$ role AdminADinfo/Headquarters
$ adinfo
```

Running Unauthorized Commands

If your administrator has assigned you to a role that requires a restricted shell environment, the dzsh shell allows you to run only the subset of commands to which you have rights. If you attempt to run a command you are not authorized to use in your current role, the shell displays a warning.

Setting or Changing your Active Role

If you are assigned only to one or more restricted shell environment roles, you are only allowed to run commands within the dzsh shell. Within the restricted shell, you can only be in one active role at a time to prevent ambiguity about the commands you can run or what account should be used to execute those commands.

For example, if you are assigned the lab_staff restricted shell environment role that specifies that the tar command should run as root, and also the temps restricted shell environment role that specifies that the tar command should be run as the account tmp_admin, you need to specify which role you are using to run the tar command under the proper account.

You can see what roles are assigned to you, as well as switch between roles, using the role command. For example, to view the list of roles to choose from, you would type:

$ role -ls

To choose the lab_staff role, you would type:

$ role lab_staff

Using PAM Application Rights

Most of the applications you run on Linux and UNIX computers are configured to use a pluggable authentication module (PAM) to control access. Secure shell (ssh), login, and file transfer (ftp) services are all examples of PAM-enabled applications.

If you have a role assignment with access to PAM-enabled application rights, you can run one or more specific applications using the administrative privileges defined for your role. The administrator defines the specific PAM application rights that you have in each role you are assigned. If you have a role assignment with application access rights, the administrator specifies the arguments you can use when running the application and the account used to run the application.

Using Secure Shell Session Rights

If your administrator has assigned you the sshd or ssh right, login-all right, or a custom PAM access right, you can use secure shell rights to perform specific operations on remote computers. The following are a list of predefined secure shell session-based rights that might be assigned to you:

- dzssh-all grants access to all available secure shell services.
- dzssh-direct-tcpip allows local and dynamic port forwarding (ssl-L, ssh -D).
- dzssh-exec allows command execution.
- dzssh-scp allows secure copy (scp) operations.
- dzsh-shell allows secure terminal (tty/pty) connections.
- dzssh-Subsystem allows external subsystems, with the exception of the sftp subsystem, which has its own right.
- dzssh-tcpip-forward allows remote port forwarding (ssh -R).
- dzssh-tunnel allows tunnel device forwarding.
- dzssh-x11-forwarding allows X11 forwarding.
- dzssh-sftp allows SSH File Transfer Protocol.

Role-based Auditing of Session Activity

Your administrator may install the Centrify Agent with or without auditing features. Depending on whether auditing features are activated on your computer and whether your role requires auditing or not, your session activity might be captured and stored in a database. You can check whether session-level or desktop auditing is requested or required for the roles you are assigned by running the dzinfo command. You are notified that your session activity might be audited only if the administrator has enabled notification. If auditing is required for your role, but the auditing service is not available on computer you attempt to use, you will be denied access to that computer until auditing is available.

If your administrator has configured the Centrify Agent to audit your session when you log in, everything you do on your terminal is captured, including all of your keystrokes and anything displayed on your screen. If your administrator has configured auditing on a per-command basis, auditing only begins when you use a privileged dzdo command, and ends when you are finished running those privileged commands.

If your administrator has configured desktop auditing, everything you do in the Linux graphical user interface is captured. Note that for web browser activity, desktop auditing captures the web page title but not the contents or activity within a web page.

Delinea

Welcome to the Delinea software User Guide for Windows.

This section provides an overview of Delinea software features for Windows computers and how you can use Delinea software to temporarily elevate your privileges to perform administrative tasks locally on your computer or remotely on a network server.

What is Server Suite?

Server Suite is a multi-tier software solution that enables administrators to centrally manage access to on-premise servers and workstation, mobile devices, and applications across a broad range of platforms. With Server Suite, administrators can accomplish the following:

- Manage local and remote access to computers with Linux, UNIX, Mac OS X, and Windows operating systems.
- Enforce security policies and control access to applications on mobile devices such as iPhone and Android smart phones and tablets.
- Enable single sign-on and role-based rights for on-site and cloud-based applications.
- Capture detailed information about user activity and the use of administrative privileges.

Using Delinea software, an Active Directory administrator creates **zones** to organize the enterprise's on-premise computers, mobile devices, and applications into groups. For each group, the administrator then defines rights, roles, and group policies to control access to the computers and applications in that zone. By using zones and role assignments, the administrator can establish fine-grain control over who is authorized to perform administrative tasks and when user activity should be audited.

With Delinea software, your organization can reduce the risk of unauthorized access to critical resources, ensure accountability and regulatory compliance for users with access to privileged accounts or sensitive information, and simplify the management of shared accounts and role-based access rights.

Using Delinea software to Manage Access to Windows Computers

Delinea provides a cross-platform solution that relies on the deployment of an Agent. To manage access to Windows servers and workstations, an administrator installs the Agent for Windows and identifies the zone the computer should use. If an administrator has installed the agent and added your computer to a zone, the computer is a **managed computer**. When you log on, the agent will check that you have been assigned a role that allows a local or remote logon. As long as you have a role assignment that allow you to log on, logging on proceeds normally. If you have not been assigned a role that allows you to log on, you will be denied access to the computer.

In most cases, an Active Directory administrator or another delegated administrator will also define rights and roles that enable you to run as another account that has elevated privileges. For example, the administrator might create a role that allows you to manage a Microsoft SQL Server instance using administrative privileges and another role that enables you to run an Exchange management tool using a shared service account.

The administrator is responsible for defining the specific rights that are available in different roles and for assigning those roles to the appropriate Active Directory users and groups. The administrator can also assign selected roles to local Windows users and groups.

As a user logging on to a **Delinea-managed computer**, you have the option to select from and switch between the roles you have been assigned. For example, you begin the day by logging on to your computer using your Active Directory credentials. In most cases, this account does not have elevated privileges. In your work queue, you find that you need to add a new database to the SQL Server instance you manage. Because this change requires administrative privileges not available in your logon account, you select the role that has elevated privileges that you have been assigned for managing SQL Server instances. When you are done adding the database in Microsoft SQL Server Management Studio, you switch back to your default logon account.

The administrator determines whether the elevated privileges in your role are limited to a specific application, for example, Microsoft SQL Server Management Studio, any application on your desktop, or only allowed on a remote server. You are responsible for selecting the appropriate role to do the work required from the list of roles available to you.

Auditing Role-based Activity

The administrator can also define an auditing requirement for each role. If you switch to a role that is audited, the switch is recorded in the local Windows event log. If the computer you are using is configured to audit session activity, all of the actions you take during the session are captured in a video recording until you end the session or log out. If session activity is audited, the agent on your computer captures everything displayed on the screen, including your keystrokes and the windows you have open while you are using an audited role on an audited computer. If you switch from a role that requires auditing to one that has no audit requirement, the recording stops until you resume the role that requires auditing.

The administrator determines which roles and computers require auditing of user activity and can enable auditing notification to inform you if your actions might be audited.

Roles Grant Different Types of Access Rights

There are three types of access rights that an administrator can add to any role you might be assigned:

Desktop	If you have been assigned a role that grants a desktop right, you can create a separate desktop on your computer to run applications as yourself but with the elevated privileges associated with a specific Active Directory or built-in group. In most cases, an administrator assigns you a role with a desktop right if you have more than one local application for which you need elevated privileges and you need to use those privileges frequently. For example, if you use several administrative applications on a daily basis, you are likely to be assigned a role that has a desktop right. **Note:** On Windows 10 and Windows Server 2016 systems, task bar menus are not available in an Elevated Desktop.
Application	If you have been assigned a role that grants an application right, you can run a specific application with the elevated privileges associated with a specific user account or as yourself but with the elevated privileges associated with a specific Active Directory or built-in group. In most cases, an administrator assigns you a role with an application right if you have only occasional administrative responsibilities for a specific application or only need temporary use of the elevated privileges.
Network access	If you have been assigned a role that grants a network access right, you can connect to a remote computer as an account with privileges on that computer. In most cases, an administrator assigns you a role with a network access right if you need to take administrative action on a remote server. This access right does not change any of your privileges on your local computer.

Every role includes one or more rights. Depending on the roles you have been assigned, you might have one or more of these access rights available.

Computers Must be in a Zone for Roles to be Available

The administrator can define different rights and different roles for every zone. Your computer must be joined to a zone for those rights and roles to be available. In addition, a computer can be joined to only one zone at a time. The rights you have in any zone are based on the roles assigned to you in that zone and its parent zone. If the administrator has not added your computer to a zone, no local or network roles will be available for you to use.

After a computer is added to a zone, it is possible that your role assignments might enable you to access remote computers in zones other than the local computer's zone. Roles that enable access to remote computers do not require you to have any local roles available in your local computer's zone.

In most cases, the administrator should add your computer to the appropriate zone. Changing the zone assignment requires local administrative privileges. If you have administrative privileges on your local computer, you can use the Privilege Elevation Service Settings to view information about your current configuration and perform administrative tasks, if required. For example, if the administrator notifies you that you should join a zone they have prepared, you can use the Privilege Elevation Service Settings to complete the operation for your local computer.

Using the dzjoin Command

The dzjoin command line program enables you to automatically join users to the zone in which their roles and rights are assigned, or to join them to a specific zone by zone name, when they log on to their computer. The dzjoin command line program is particularly useful for organizations that use non-persistent virtual desktop infrastructures.

The syntax for the dzjoin command is:

dzjoin [/c ‹domain controller›] [/d] [/u ‹username›] [/f] [/h] [/r [ylnlyeslno]] {/z ‹zonename› I /s I /v]

Note: {/b}If the u option is specified but no password is found in the redirected input, you will be prompted for a password.

/c	Specify a domain controller to connect to.
/d	Retrieve zone data before restarting
/u	Specify the user name to join zone using custom credentials. The user name must be in the format: USER@DOMAIN or DOMAIN\USER. The credentials are for remote access only. For the password, you can specify by redirected input. Otherwise, this tool will prompt user for password.

/f	Suppress any warnings and/or questions.
/h	Displays the command help.
/r	Suppress the restart warning and specify to restart machine, if required, after joining zone. If no restart is required, this option is ignored. If no argument is provided, e.g. '/r', the default is to restart (example: '/r yes').
/z	Join a zone using the zone name. If the zone name is not unique, use the canonical name instead.
/s	Join to the zone where this computer is already pre-created in the zone or had previously been joined to the zone (but remotely left in a disconnected situation).
/v	Display the agent version.

Note: {/b}You can also use the PowerShell command Join-CdmZone to join a zone.

Using the dzleave Command

To leave a zone, use the dzleave command. The syntax for the dzleave command is:

dzleave [/c <domain controller>] [/u <username>] [/a|/f] [/r [y|n|yes|no]] [/v] [/h]

/a	Remove the role assignment from the computer zone.
/c	Specify a domain controller to connect to.
/u	Specify the user name to leave zone using custom credentials. The user name must be in the format: USER@DOMAIN or DOMAIN\USER. The credentials are for remote access only. For the password, you can specify by redirected input. Otherwise, this tool will prompt user for password.
/f	Suppress any warning and/or question(s). In case the domain cannot be contacted, this tool will perform a local zone leave automatically.
/h	Displays the command help.
/r	Specify whether to restart machine, if required, after leaving zone without prompt. If no restart is needed, this option is ignored. If no argument is provided, example: '/r', the default is to restart ('/r yes').
/v	Show the agent version.

Note: You can also use the PowerShell command Exit-CdmZone to leave a zone.

Why You Should Use Roles for Administrative Tasks

Roles give the administrator complete flexibility for delegating control and limiting risk. For example, the administrator can define a role that lets you do specific administrative functions on your local or a remote computer without giving you the administrator's password. By eliminating the use of a shared password for the administrator's account, you can prevent an audit finding that could be costly for your organization. Using a role also limits your authority on the computer, ensuring appropriate accountability, and limits the potential damage a compromised password might cause.

In addition, roles enable targeted auditing of user activity, so that only the actions when you have elevated privileges or access certain computers are recorded. In many cases, these activities must be recorded for regulatory or industry compliance. With roles, you can go about your normal activity, such as

reading and responding to email, without auditing, then capture detailed information about the use of SQL Server Management Studio or the Exchange Management Console.

What Gets Installed on a Managed Computer

The Agent for Windows package contains software to support auditing, access control, and privilege management on Windows computers. These features must be installed together on any supported Windows computer. Depending on the services to be enabled, your computer might include the following:

- Privilege Elevation Service manages your access rights, including your ability to log on locally, connect to a remote server, and access applications using administrative privileges.
- Privilege Elevation Service desktop applet that enables you to select roles, open new desktops, switch between open desktops, and view details about our role assignments. The applet is visible on your computer as the Delinea icon in the system tray.
- Privilege Elevation Service Settings that enable an administrator to join, change, or leave the zone, run diagnostics, and configure and view logged activity.
- Identity Platform Settings that enable multi-factor authentication (MFA) login, enable RADIUS authentication, and other identity services.

If you are assigned roles that define application and desktop rights on your local computer, or access rights on remote computers, the Agent for Windows must be installed on your local computer and on the remote computer.

The administrator can deploy the Agent for Windows from a central location on the network to your computer or you can install it directly on your local computer.

This section includes information on how to use Delinea software to access applications with privileges on a Windows computer that has the Agent for Windows installed.

Verify That You Can Log On

The Agent for Windows can be centrally deployed by a system administrator or deployed locally directly on a computer. If an administrator has installed a Agent on a computer you use, the next step is to verify that you can log on successfully and locate the Delinea applet on your computer. The Agent does not change how you log on your computer. However, you must be assigned at least one role that allows you to log on locally, remotely, or both.

When you are prompted for a user name and password, type your domain or local credentials as you normally would. If your administrator has enabled multi-factor authentication for log in, you will be asked to perform one or more authentication challenges, such as responding to a text message or email message, answering a security question, or answering an automated phone call. If you provide valid credentials and have been assigned a role with permission to log on, you should see your default desktop as it normally displays with the addition of a Delinea applet that is added to the system tray notification area. By left-clicking on the Delinea applet, you can view your current desktop and assigned roles.

As part of the deployment, your computer may or may not have been joined to a zone. If the administrator has not specified a zone for your computer to join as part of the deployment process, you can specify a zone using the Privilege Elevation Service on your local computer. Contact your system administrator to find out which zone you should join.

You can check whether the agent is installed and running, and whether you are connected to a zone using the Services Control Panel or the Privilege Elevation Service. For example, click **Start > All Programs > Server Suite 2021.1 > Agent Configuration > Privilege Elevation Service** to view the control panel.

If the agent is installed but not connected to a zone, you should contact your system administrator to determine the zone to use. You should note, however, that setting or changing the zone assignment requires local administrative privileges. If the agent is not installed on the local computer, you should contact your administrator to find out if you are responsible for deploying the Agent for Windows on your computer.

If the zone information for the agent is configured, but the agent status is not Connected, your current rights, roles, and role assignment privileges should still be available, in most cases, from the local authorization cache. If you are unable to perform administrative tasks that you normally can perform, contact your system administrator to determine whether the authorization cache needs to be refreshed.

If you cannot log on, see What to do if you cannot log on.

What to Do if the Delinea Icon is Not Displayed

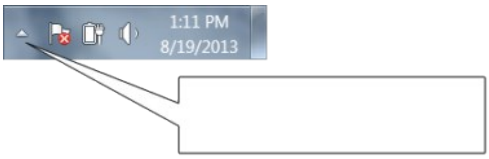
By default, the Windows system tray notification area is the located to the right of the task bar on the bottom of your screen. You can customize this area to display icons for different applications.

If the Delinea icon is not displayed by default, you can click the up arrow in the notification area to add it.

To display the Delinea icon if it is not displayed by default:

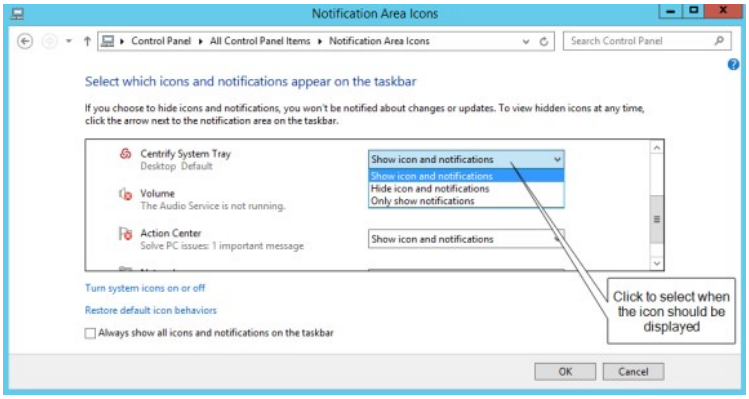
1. Click the up arrow in the notification area.

   For example:

2. Click **Customize** to change the icons displayed.
3. Scroll to the System Tray icon, then select **Show icon and notifications** to display the Delinea icon at all times.

   For example:



4. Click **OK**.

What to Do if You Cannot Log On

There are several reasons why an attempt to log on can fail. If you are denied access to a computer:

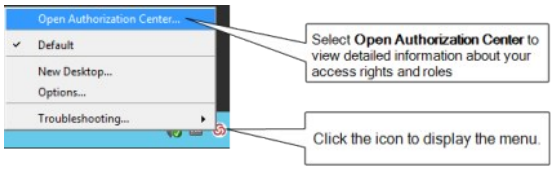
- Verify that the computer you trying to log on to allows the type of access you are attempting. For example, most users cannot log on locally on computers that are Active Directory domain controllers. Similarly, a computer's properties must be configured to allow remote access for you to be able to connect remotely. These settings are Windows policies and properties and are not related to the Agent for Windows.
- Check whether you are attempting to log on using a local account or a domain account. The administrator can assign a role that allows you to log on to your local account, your domain account, or both. It is possible that only one of those accounts has been assigned a role with access to the computer. For example, your administrator may have your account configured so you can log on using your local account credentials but not with your domain credentials.
- Verify that the computer where you are trying to log on has access to an Active Directory domain controller. If an Active Directory domain controller is not available or the local computer is not a member of an Active Directory domain, you might be prevented from logging on because the agent cannot verify you have authority to access the computer.
- Determine whether you are attempting to log on to a remote computer with an appropriate role. The administrator can assign a role that allows you to log on locally, log on remotely, or both. It is possible that only one of those rights has been configured for the role you have been assigned. For example, your administrator may have configured the role you are assigned to allow you to log on to your local computer but not allow remote connections.

After the Agent for Windows has been installed, you must have a role assigned to your account that gives you log on privileges. If an attempt to log on fails, contact your Active Directory administrator or helpdesk to determine the roles you have been assigned, the type of access your roles grant, and any limitations associated with your role assignment. For example, roles can have time constraints with specific periods of availability. If you attempt to log on, but the role is not available, you will be denied access.

For more information about the steps you can take, see Troubleshooting.

Checking Your Rights and Role Assignments

The roles you are assigned control your access rights and the accounts you can use to log on. You can look up detailed information about your rights and role assignments by right-clicking the Delinea icon, then selecting Open Authorization Center. For example:



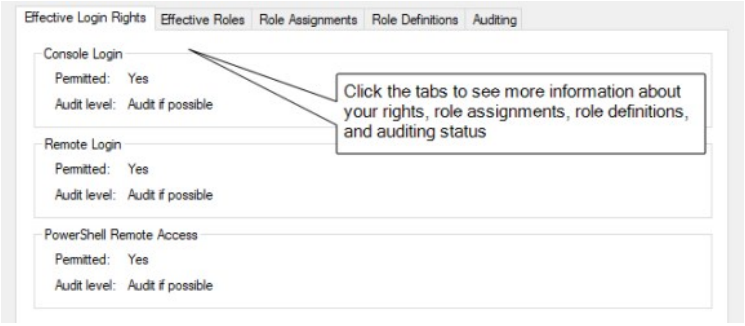
You can then click the tabs to see information about your current role and any other roles you have been assigned. For example, from Authorization Center, click the following tabs to see more detailed information about how the roles you have been assigned are configured:

- Click **Effective Login Rights** to see information about your local, remote, and PowerShell login rights and whether auditing is requested, required, or not applicable.
- Click **Effective Roles** to see information about the roles you have been assigned and the current status of each role. For any roles, you can right-click a

role, then select Role Properties to view additional details. For example, if any of your roles are Inactive, you can right-click to see the time constraints defined for the role. You can also view the specific type of rights granted by each role.

- Click **Role Assignments** to see detailed information about your role assignments, including where the assignment was made, whether the role is a local or network role, and the start and end times that are in effect for the role. You can right-click a role assignment, then select Assignment Properties or Role Properties to view additional details.
- Click **Role Definitions** to see detailed information about the login rights and audit requirements that have been defined for the roles you have been assigned. You can rightclick a role definition, then select Properties to view additional details.
- Click **Auditing** to see information about the auditing status for each desktop started in a session.

You can only view information about your own access rights and role assignments in the Authorization Center. Click **Close** when you are finished viewing authorization information.



After you review information about your access rights and role assignments using Authorization Center, you should have a basic understanding of the roles you have been assigned, any restrictions on when they are available, and what the roles allow you to do. Your role assignments control where you can log on, the type of account you use to log on, the specific access rights you have on local or network computers. As discussed in Roles grant different types of access rights, there are three categories of access rights for Windows computers:

- Desktop
- Application
- Network access

Depending on the details of how roles are defined in your organization and the specific roles you have been assigned, you might have some or all of the access rights described in the next sections.

Working with Desktop Access Rights

When you first log on, the default desktop is your only desktop. Depending on whether you logged on using a local user account or an Active Directory domain user account, you have the default privileges associated with that account. If you have been assigned a role with a desktop access right, the Agent enables you to run individual applications using a selected role from your default desktop or create one or more new desktops to run multiple applications using the administrative privileges associated with your roles.

If you have one or more roles with desktop rights, you can create, select, and switch between desktops on computers that have a traditional Windows desktop.

> **Note**: If the computer you are using is running Windows 8 or 8.1, or Windows Server 2012 or 2012 R2, Windows does not provide access to applications natively when you switch from the default desktop to a privileged desktop due to changes to the underlying interfaces and supported features within the operating system. To enable access to applications on computers running these versions of Windows, the Agent for Windows provides a custom start menu. The start menu allows you to open and run applications as you would on Windows 7 or Windows Server 2008 R2. The start menu is installed on the left side of the taskbar and displays the Delinea logo. This start menu is only available if you are using a role with Delinea desktop rights and cannot be modified.
>
> **Note**: If you launch a Universal Windows Platform (UWP) application in the default desktop, there won't be a response if you click the same application in the privileged desktop. You need to close the application in the default desktop before you can open it in the privileged desktop.

Running an Individual Application Using a Role

If you have a role assignment with a desktop access right, you don't have to create a new desktop to run a local application using your administrative privileges. You can select any local application directly from your default desktop, then select a role you have been assigned without creating a new desktop

or switching from one desktop to another. This is often the best solution if you only run one application using your administrative privileges or rarely need to invoke those privileges.

To run a local application using a selected role:

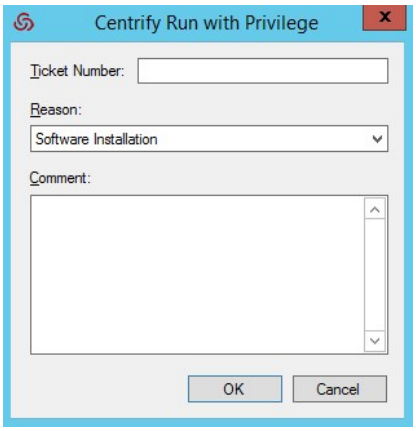
1. Navigate to and select the application you want to run.
2. Right-click the executable or shortcut for the application.

   If you want to open the application from the Start menu, press the Shift key when you right-click.
3. Select **Run with Privilege**.

   Selecting **Run with Privilege** is similar to selecting standard Windows "Run as" or "Run as administrator" menu items, but does not require you to provide a password for an administrative or shared service account. Instead, you always use your own password to authenticate your identity.
4. If the Select Role dialog box opens, select a role from the list of available roles, then click **OK**.

   > **Note**: If there is only one role assigned to you that allows you to run the application, the application will automatically run using that role, and the dialog box does not open. If you would like to access the Select Role dialog box, press the Shift key when you select **Run with Privilege**.
5. Type the password for your login account if you are prompted for it, then click **OK**.

   If your administrator has enabled privilege elevation justification, a justification dialog box appears.
6. Enter the following information to justify why you need to run the application with privilege:



   - **Ticket number**: If your administrator has instructed you to enter a ticket number, do so here. (This field can be used with ticketing systems such as ServiceNow and so forth.)
   - **Reason**: Select the reason category that best fits your situation. Your choices are:
     - Software Installation
     - Remote System Administration
     - Local System Administration
     - Windows Feature Management
     - System Networking Change
     - Maintenance (Shutdown, Reboot, Power Off)
     - PowerShell or Other CLI
     - Operation (Services, Zone Operations, etc.)
     - Other
   - **Comment**: Enter any comments about your need to run with privilege.
7. If your administrator has enabled multi-factor authentication, complete the additional authentication challenges after entering your password.

   After you select a role, you have the rights associated with that role. The application opens with the privileges associated with a specific user account or with the members of a particular administrative group and an audit trail event is recorded in the Windows Application event log. When you close the application, you resume working with your normal account privileges and group membership.

Creating a New Desktop

Desktop access rights enable you to create a separate desktop working environment for each role the administrator has assigned to you. You might have multiple role assignments with different desktop access rights so that you can run applications with elevated privileges. For example, you might be assigned two separate roles—one for running applications as a member of the domain administrators group and another for running applications as a member of the local administrators group.

If you have been assigned roles that have desktop access rights, you can create a desktop for each role.

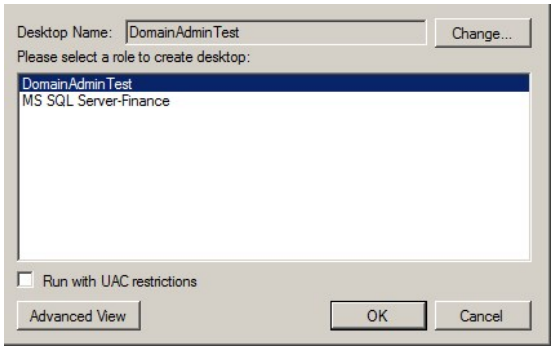
To create a new desktop:

1. Click the Delinea icon in the notification area.
2. Select **New Desktop**.

   If you have not been assigned to any role that has a desktop access right, a message is displayed to inform you that you are not a member of any role that permits opening a new desktop.

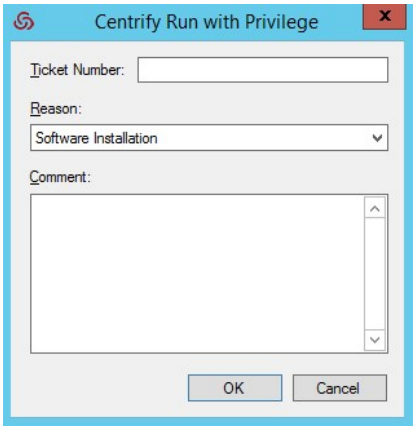
   If you have been assigned to any roles that have desktop access rights, you can continue to the next step.
3. Select a role from the list of your available roles, then click **OK**.

   For example, if you are assigned multiple roles that include desktop access rights, you can select from these role assignments to control which account privileges are in effect for the new desktop.



   Note that the roles listed might allow you to run as your own account locally, but grant access to remote servers. To see more information about the context associated with your roles, click **Advanced View**.

   When you select a role, you also have the option to run the desktop with User Account Control (UAC) restrictions enforced. Selecting this option gives you filtered privileges, prompting you to confirm actions before continuing with operations that require elevated privileges. You can leave this option unselected to use a desktop with full privileges and without being prompted to confirm your actions. You should note, however, that when you run a desktop without enforcing UAC restrictions, no warnings are displayed, even if you have configured User Account Control Settings on the local computer.
4. Type the password for your login account, if you are prompted for it, then click **OK**.
5. If your administrator has enabled privilege elevation justification, a dialog box appears. Enter the following information to justify why you need to run the application with privilege:



- **Ticket number**: If your administrator has instructed you to enter a ticket number, do so here. (This field can be used with ticketing systems such as ServiceNow and so forth.)
- **Reason**: Select the reason category that best fits your situation. Your choices are:
  - Software Installation
  - Remote System Administration
  - Local System Administration
  - Windows Feature Management
  - System Networking Change
  - Maintenance (Shutdown, Reboot, Power Off)
  - PowerShell or Other CLI
  - Operation (Services, Zone Operations, etc.)
  - Other
- **Comment**: Enter any comments about your need to create a new desktop.

6. If your administrator has enabled multi-factor authentication, complete the additional authentication challenges after entering your password.
7. After you select a role and click **OK**, the new desktop becomes your working environment. You can view the local and network roles you are using for the new desktop by left-clicking on the Delinea icon in the system Notification Area on the taskbar.

   If the role is only applicable on a remote computer, the local role is displayed as Self. If the role does not have network access rights, the network role is displayed as Self.

   To see complete information about the desktop, application, and network access rights for each of your roles, open the Authorization Center as described in Checking your rights and role assignments.

Setting a Desktop Name

By default, new desktops uses the name of the role you select as the desktop name. You can click **Change** if you want to change the name of desktop. For example, you might want to add your name, a computer name, or other information to the information displayed when you left-click the Delinea icon in your system Notification Area to help identify the context when switching from one desktop to another.

After you click **Change**, select **Use the following desktop name**, type the name you want displayed for the desktop, then click **OK**.

Switching from one desktop to another

To switch desktops, click the Delinea icon and select the desktop you want. You can also set up hot keys to switch between desktops using a keystroke combination.

Setting hot keys for switching between desktops

Hot keys are keystroke combinations that enable you to switch between desktops without clicking the Delinea icon or accessing the applet menu. By selecting hot key combinations, you can move from one desktop to another more quickly when you have more than one desktop open at a time.

To set up hot keys for switching between desktops:

1. Click the Delinea icon.

2. Select **Options**.
3. Click the **Hotkey** tab.
4. Select the **Enable hotkey** option, then select a key or key combination from the list of Modifiers and whether to use a number, function key, or letter from the list of Specifiers.

   For example, if you want to switch from one desktop to another using the Alt and a number, Alt+1 and Alt+2, select Alt from the list of Modifiers and Number from the list of Specifiers.
5. Click **OK**.

Using a desktop with network access rights

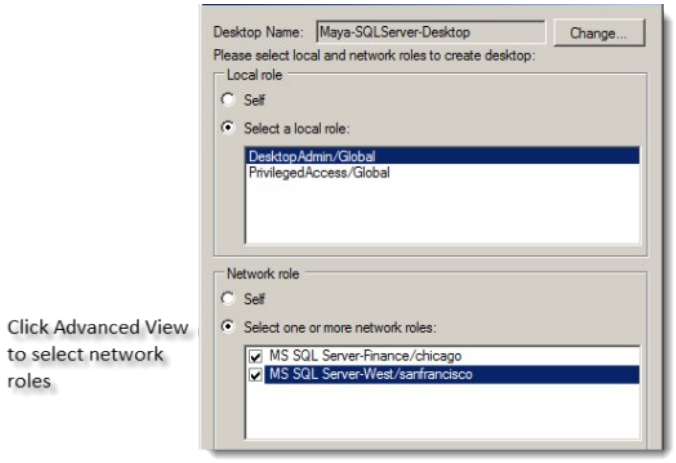
When you open a desktop and select a role, you get all of the access rights associated with that role. Depending on how the role is configured, those access rights may be limited to running applications with locally elevated privileges or include access to remote servers on the network. The Delinea icon in the system Notification Area always displays the current Local and Network roles you are using. However, it is up to the administrator to decide whether network access rights should be included in roles that grant desktop access rights.

If roles granting network access rights are defined separately from roles that include desktop access rights, you might have to select your local and network roles separately. In some cases, you might also need to select more than one network role to work with multiple remote computers. To handle these more complex situations, you can use the Advanced View to select the appropriate combination of local and network roles.

To view and select your local and network roles for a desktop:

1. Open a new desktop.
2. In the Select Role dialog box, click **Advanced View**.

   If there are any network roles listed, those roles grant network access rights for specific remote computers. For example, if you are assigned separate roles with network access rights to two separate SQL Server instances, you might see the roles with network access rights listed separately from your roles with local desktop access rights.



   In this example, the DesktopAdmin role is a local role that has desktop access rights but does not include any network access rights. By selecting both MS SQL Server-Finance and MS SQL ServerWest network roles, you can create a single local desktop that has remote network access to both SQL Server instances. Alternatively, you could create separate desktops for accessing each SQL Server instance. You can left-click on the Delinea icon in the system Notification Area to view the roles you have selected so that you know whether you have network access rights for one SQL Server instance or both.
3. For the local role, select a role that grants desktop access rights or application access rights on the local computer.
4. Type the password for your login account, if you are prompted for it, then click **OK**.

   If your administrator has enabled multi-factor authentication, complete the additional authentication challenges after entering your password.

Closing a desktop

In some cases, you might have multiple desktops open at the same time to allow you to switch between several different roles quickly. If you have more than one desktop open at a time, you can selectively close the desktops you are no longer using.

To close a desktop:

1. Switch to the desktop you want to close.
2. Click the Delinea icon.
3. Select **Close Desktop**.

The agent removes that desktop from your list and returns you to the default desktop. You cannot close your default desktop.

Running a specific application with privileges

With desktop access rights, you can run any application using one of the roles assigned to you. Application access rights are assigned on an application-by-application basis.

If you have a role assignment with application access rights, you can run one or more specific applications using the administrative privileges defined for your role. The administrator defines the specific application rights that you have in each role you are assigned. If you have a role assignment with application access rights, the administrator specifies the location of the application executable, the arguments you can use when running the application, and the account used when you run application. You can only select a role to run a local application for which you have application rights.

Selecting **Run with Privilege** is similar to selecting standard Windows "Run as" or "Run as administrator" menu items, but does not require you to provide a password for an administrative or shared service account. Instead, you always use your own password to authenticate your identity.

For information about running an application as an alternate user, see Run with privilege as an alternate user.

To run a local application using a selected role:

1. Navigate to and select the application you want to run.
2. Right-click the executable or shortcut for the application and select **Run with Privilege.**

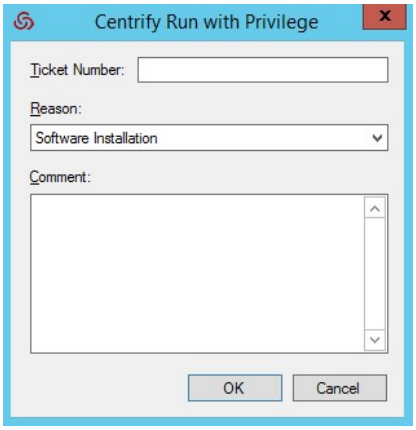
   (If you want to open the application from the Start menu, press the Shift key when you right-click.)

   If you have not been assigned to any role that has application access rights for the application you are trying to open, a message displays to inform you that you are not a member of any role associated with the selected application.

   The **Run with Privilege** dialog box displays. (If it doesn't display, it's because you're assigned to just one role, so there's no need to select a role.)

   Note: {/b}Note: If you pressed the Shift key when you right-clicked the application in Step 2, the **Run with Privilege** dialog box displays even if you're assigned to just no roles or just one role for access to that application.
3. Select the desired role.
4. If the application requires network access rights for a remote server, click **Advanced View** to see if you have a role with network access rights available.
5. If you'd prefer to use your environment variables instead of the variables that are associated with the selected role, select **Use current environment variables instead of "Run As" user's**.
6. Click **OK** to continue.
7. Enter the password for your login account, if you are prompted for it, then click **OK**.

   If your administrator has enabled privilege elevation justification, a dialog box appears.
8. Enter the following information to justify why you need to run the application with privilege:



- **Ticket number**: If your administrator has instructed you to enter a ticket number, do so here. (This field can be used with ticketing systems such as ServiceNow and so forth.)
- **Reason**: Select the reason category that best fits your situation. Your choices are:
  - Software Installation
  - Remote System Administration
  - Local System Administration
  - Windows Feature Management
  - System Networking Change
  - Maintenance (Shutdown, Reboot, Power Off)
  - PowerShell or Other CLI
  - Operation (Services, Zone Operations, etc.)
  - Other
- **Comment**: Enter any comments about your need to run with privilege.

9. If your administrator has enabled multi-factor authentication, complete the additional authentication challenge.
10. Click **OK**.

    After you've successfully authenticated, the application opens and an audit trail event is recorded in the Windows Application event log. You can use the application with the privileges granted to the specific user account or administrative group defined for your role. You have the privileges associated with the role or roles you selected until you exit the application. When you close the application, you resume working with your normal account privileges and group membership.

Using the runasrole command line

As an alternative to selecting Run with Privilege from the right-click menu for an application, you can use the runasrole command-line program. The RunAsRole program enables you to run a specified Windows application in a Command Prompt windows using a specified access role. You can use command line options to control whether the role is used as a local role, a network role, or both, and whether to use the current environment or the environment variables associated with the "Run As" user account. The runasrole command line program is equivalent to selecting the Run with Privilege menu option when right-clicking an application shortcut or executable.

The syntax for the runasrole command is:

runasrole /role:role[/zone] [options] application [argument]

runasrole /localrole:role[/zone] [options] application [argument]

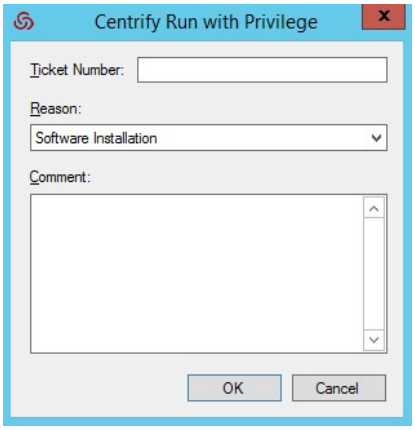
runasrole /networkrole:role[/zone] [options] application [argument]

You must specify the role to use in the rolename/zonename format. You must also specify an appropriate path to the application you want to access, including any required or optional arguments.

You can use the following command line arguments with the runasrole command:

/role	Use the role name you specify as both a local role and a network role. You can specify this option to run an application locally and access a remote server using the same role, if applicable. You should only use this option if the role you are assigned and want to use has both local and network access rights defined.
/localrole	Use the role name you specify as a local role.
/networkrole	Use the role name you specify as a network role.
/env	Use the current environment variables instead of the environment variables associated with the "Run As" user account.
/netdrives	Use mapped network drives when running an application with the selected role. By default, you cannot use mapped network drives that are associated with you logged-on user account when running applications using a role with elevated privileges. If you want to use a mapped network drive when accessing an application using a selected role, include the /netdrives option in the command line.
/removetimestamp	Remove the grace period on Windows authentication and MFA for the current user session.
/wait	Prevents the runasrole program from exiting immediately after opening the specified application. If you specify this option, the runasrole program starts the specified application and waits until the application session ends before exiting. When the application session ends, the runasrole program exits and returns the same result code as the application. If you specify this option and the application is a command line utility, the runasrole program redirects the application's input and output to the command line console. You should note that some applications use a Microsoft API that does not support redirection of standard input and output. For applications that don't support redirection, the /wait option has no effect and is ignored.
/h	Displays the command help.

Note: If your administrator has enabled privilege elevation justification, a dialog box appears. Enter the following information to justify why you need to run the application with privilege:



- **Ticket number**: If your administrator has instructed you to enter a ticket number, do so here. (This field can be used with ticketing systems such as ServiceNow and so forth.)
- **Reason**: Select the reason category that best fits your situation. Your choices are:
  - Software Installation
  - Remote System Administration
  - Local System Administration
  - Windows Feature Management
  - System Networking Change
  - Maintenance (Shutdown, Reboot, Power Off)
  - PowerShell or Other CLI
  - Operation (Services, Zone Operations, etc.)

  - Other
- **Comment**: Enter any comments about your need to run as role.

Examples of using runasrole

To use the same role to open the Computer Management application locally and access a remote server in zone1, you might run a command similar to the following:

runasrole /role:role1/zone1 mmc.exe c:\windows\system32\compmgmt.msc

To use the role named SQLdba from the finance zone as a local role to open the Services application, you might run a command similar to the following:

runasrole /localrole:SQLdba/finance mmc.exe c:\windows\system32\services.msc

To use role1 from zone1 as a local role to open the Computer Management application and use network access rights from role2 in zone2, you might run a command similar to the following:

runasrole /localrole:role1/zone1 /networkrole:role2/zone2 mmc.exe compmgmt.msc

To open the Services application using the role named SQLdba from the finance zone and have the runasrole program remain open until you close the Services application, you might run a command similar to the following:

runasrole /wait /role:SQLdba/finance mmc.exe c:\windows\system32\services.msc

Running an application from a shortcut

In most cases, you can use the runasrole program to run specified Windows applications using the application shortcut. However, there are many different types of application shortcuts and the RunAsRole program does not support all of them. You can use the RunAsRole program to execute applications with the following recognized shortcut target extensions:

.bat

.cmd

.cpl

.exe

.msc

.msi

.msp

.ps1

.vbs

.wsf

How to determine whether RunAsRole supports an application shortcut

You can determine whether you can use the RunAsRole program to execute an application from the application shortcut by checking the file extension for the target application in the application's shortcut properties dialog box.

To check the file extension for a target application shortcut

1. Select an application shortcut.
2. Right-click the shortcut, then click **Properties** to display the file properties.
3. Click the Shortcut tab and check the target field.

   If the target file extension displayed is a supported file extension, you can use RunAsRole to execute the application from the application shortcut. You should note that a shortcut target field might include both the file name for the application executable and one or more arguments. As long as the application executable has a supported file extension, you can use RunAsRole to execute the application with the specified arguments from the

shortcut. For example, if the shortcut target is C:\Windows\System32\control.exe printers, the application executable C:\Windows\System32\control.exe is a supported file extension with printers supplied as an argument. Therefore, you would be able use RunAsRole to run the application from its shortcut.

Run with privilege as an alternate user

If your administrator has enabled the group policy for this feature and assigned you to a role with rights for a specific application, you can run the application with elevated privileges of an alternate user.

For example, if your user account doesn't have privileges to install an application on the computer but your administrator does, you can stay logged in and run the application with privilege and your administrator can enter her credentials so that you can install the application.

To run a local application as an alternate user:

1. Navigate to and select the application you want to run.
2. Right-click the executable or shortcut for the application and select **Run with Privilege.**

   (If you want to open the application from the Start menu, press the Shift key when you right-click.)

   If you have not been assigned to any role that has application access rights for the application you are trying to open, a message displays to inform you that you are not a member of any role associated with the selected application.

   The **Run with Privilege** dialog box displays. (If it doesn't display, it's because you're assigned to just one role, so there's no need to select a role.)

   Note: {/b}Note: If you pressed the Shift key when you right-clicked the application in Step 2, the **Run with Privilege** dialog box displays even if you're assigned to no roles or just one role for access to that application.
3. To specify the alternate user, click **Change User** and specify the alternate user account.
4. When prompted, enter the alternate user's login credentials to authenticate as that user.

   The dialog box now lists the alternate user and the roles assigned to that account.
5. Select the desired role.
6. If your administrator has configured the system to re-authenticate, enter the alternate user's credentials again and click **OK**.
7. If your administrator has enabled multi-factor authentication, complete the additional authentication challenge.
8. If you'd prefer to use your environment variables instead of the variables that are associated with the alternate user, select **Use current environment variables instead of "Run As" user's**.

   For example, this option is useful if you're installing a program that's dependent on one that you've installed and is set in your path variable
9. Click **OK**.

   After you've successfully authenticated with the alternate user's credentials, the application opens and an audit trail event is recorded in the Windows Application event log. You can use the application with the privileges granted to the specific user account. You have the privileges associated with the specified user account until you exit the application.

Running an application with an alternate account

If your administrator has configured the ability to run an application with an alternate account, you can run applications with an alternate, privileged account without having to log in to PAS and check out the password for that alternate account.

Alternate accounts are typically a privileged or administrator account in Active Directory that's associated with an owner account. You can log in to the alternate account using your main account.

For example, system administrators typically have several accounts, a user account for general log-ins and an administrative account to access specific systems and services.

To run an application with an alternate account:

1. Right-click the desired application and choose Run with Alternate Account.

2. If you have multiple alternate accounts, you can then choose which account to use.

   The application runs under your alternate account.

Selecting roles with network access rights

As discussed in Using a desktop with network access rights network access rights can be included in roles with other rights or defined separately. Therefore, it is not always possible to see where your rights apply or the scope of your role assignment.

If you are assigned multiple roles, you should work with the administrator to identify which roles grant local and network access rights and the computers where the roles apply. You can see detailed information about the rights associated with each role you are assigned and the zones where different roles are defined using the Authorization Center. You have less visibility, however, of which computers are in scope for your network access rights.

Selecting a role that is not applicable on a local computer

In some cases, you might have roles that are visible on your local computer in the list of roles you have been assigned that are not applicable on the local computer. You can select the role, but the privileges associated with the role are only granted when you access computers over the network where the assignment applies.

For example, an administrator might create an Exchange Admin role that contains a network access right, and assign you to that role in a zone that only contains Exchange servers or assign you to that role explicitly on the computers that host Exchange.

When you log on to your laptop, the Exchange Admin role is included in your list of available roles even though the assignment is out of scope for the laptop. You can select the Exchange Admin role and continue working on the laptop without elevated privileges.You know that the Exchange server requires maintenance and you are planning to get to it later in the day.

When you are ready to do maintenance on the Exchange server, you connect to the server over the network. At that point, the elevated privileges associated with the Exchange Admin role are applied. The Exchange server you are accessing from your laptop is in scope for where you have been assigned the Exchange Admin role. You complete the maintenance required on the Exchange server with your elevated privileges, then resume working on your laptop where the Exchange Admin role does not apply.

Role-based auditing of session activity

The Agent for Windows can be installed with or without auditing features. Depending on whether auditing features are activated on your computer and whether your role requires auditing or not, your session activity might be captured and stored in a database. You can check whether session-level auditing is requested or required for the roles you are assigned using Authorization Center. You are only notified that your session activity might be audited if the administrator has enabled notification. If you select a role that requires auditing but auditing features are not available on computer you attempt to use, you will be denied access to that computer until auditing is available.

If session-level auditing is activated, everything you do on your computer is captured, including all of your keystrokes and the screens displayed on the desktops you use. At a minimum, any time you use a role that elevates your privileges on a computer, an audit trail event is recorded in the Windows Application event log.

Setting up the offline MFA profile (multi-factor authentication)

If you are required to use multi-factor authentication, you may be prompted to set up an offline MFA profile so that you can access your computer in the event that the authentication server cannot be reached.

Note: {/b}If you have already set up your offline MFA profile and want to reconfigure (override) it, you will be prompted for multi-factor authentication. That profile is set in the MFA Login Policy.

If your administrator has enabled offline multi-factor authentication, you will see a notification message each time you log on which will prompt you to set up your offline MFA profile. Depending on the configuration settings, you may not be able to access your machine in the event that you are unable to connect to the authentication server if you do not set up the offline MFA profile.

To set up an offline MFA profile:

1. Right click the notification icon in the system notification area, and select **Setup Offline MFA Profile**.
2. Click **Next** to begin the Offline Authentication Wizard.
3. Select one of the following methods to create an authenticator account profile on your mobile device:
   - **Scan barcode**

If you select this option, a QR code is displayed for you to scan using your mobile authenticator application. You can use either the Delinea application or a third-party authenticator application.

- **Manual entry**

  If you select this option, you must manually enter the displayed account profile information into your authenticator application.

- **Program YubiKey**

  If you select this option, you can use a YubiKey as the second form of authentication. You'll then need to select which slot on the YubiKey to use, and whether or not to use Yubikey's touch-to-sign feature.

4. Enter the passcode that is generated after you have created your authenticator profile. Click **Next**.
5. Click **Finish** to exit the Wizard.

After you have set up your offline MFA profile, you will be prompted to enter the mobile passcode generated by your authentication application as the second form of authentication when you attempt to log on to your machine if it cannot connect to the authentication server.

Authentication grace periods

When you have authenticated with a Delinea software component either with Windows authentication or MFA, you have a short period of time where you won't need to re-authenticate for the same type of item.

Understand that there are 3 types of grace periods for authentication:

- Lock Screen MFA grace period
- User Privilege Elevation for MFA grace period
- User Privilege Elevation for Windows Authentication grace period

Your administrator enables and configures these grace periods by way of a group policy, and each grace period type has its own policy. By default, these grace periods are not in effect.

For the lock screen MFA grace period, when you lock the screen within the grace period (either you lock the screen yourself or if your screen saver does it for you), you can unlock the login session without an MFA challenge.

If the group policy "Continue with MFA Challenges after failed windows authentication in Logon Screen" is enabled, then the lock screen MFA grace period is disabled automatically.

For the user privilege elevation grace period (MFA or Windows authentication) , the grace period is triggered when you either run an application with privilege, switch to a privileged desktop, or create a new privileged desktop. During the grace period, you aren't requested to re-authenticate by way of MFA or Windows authentication, respectively.

For both the user privilege elevation grace periods (MFA and Windows authentication), you can clear the grace period manually. To clear the grace period, right-click the Delinea icon in the system tray and select **Clear Grace Period** › **MFA or Clear Grace Period** › **Windows Authentication**. The Clear Grace Period option is only enabled if you're within the user privilege elevation grace period.

Agents can be installed on Windows computers that are configured to run the Server Core operating environment. Server Core is a Windows installation option that provides a low-maintenance server environment with limited functionality.

Most Agent operations are not affected by running on Server Core. However, there are specific features that are not available or not applicable because of the limitations of the Server Core environment itself. For example, the Run with Privilege menu option is not available on Server Core computers because Server Core does not support Windows Explorer and other graphical user interface applications. However, you can use the runasrole command line utility to run specific applications using a specified role.

Similarly, there's no notification area applet or desktop rights available on Server Core computers. However, you can access the Authorization Center, agent control panels, and agent command-line utilities from the Server Core command prompt.

The following list summarizes the Agent for Windows features that are not supported on Server Core computers:

- You cannot create, select, or switch desktops or use any desktop-related features because the Windows desktop is not available on Server Core.
- You cannot select Run with Privilege as a right-click menu option for applications because Windows Explorer is not available on Server Core.
- You cannot open the Authorization Center or access the notification area applet because the Windows desktop and Windows Explorer are not available on Server Core.
- You cannot open applications such as the Privilege Elevation Service Settings or DirectAudit Agent Control Panel from Start menu shortcuts because the Windows desktop and Windows Explorer are not available on Server Core.

You should note that only Agents for Windows are supported for the Server Core environment. A small number of other Server Suite for Windows support a command line interface, but are not configured to support a Server Core environment.

Server Core supported platforms

Delinea supports the following versions of the Server Core environment:

- Windows Server 2012 Server Core
- Windows Server 2012 Minimal Server Interface
- Windows Server 2012 R2 Server Core
- Windows Server 2012 R2 Minimal Server Interface

You should note that Server Core is not supported on Windows Server 2008 because Windows Server 2008 Server Core does not support any version of the .NET Framework. The Agent for Windows requires the .NET Framework. For more information about the supported libraries and .NET functionality on Server Core, see the reference material available on the Microsoft Developer Network website for the operating system you have deployed.

Joining a zone

One of the first tasks after installing the Agent is to join a zone. You can do by launching the Privilege Elevation Service Settings from the command prompt.

To open the Privilege Elevation Service Settings to join a zone:

1. Navigate to the Agent installation directory.

   By default, the agent files are installed in the C:\Program Files\Centrify\Agent for Windows directory.

2. Run Centrify.DirectAuthorize.Agent.Config.exe.

3. Click **Join zone**.

4. Type all or part of the zone name, click Find Now, then select the zone to join and click **OK**.

5. Click **Close** to close the control panel.

If you later need to change the zone, run diagnostics, refresh the authorization cache, or view or modify log settings, you can run Centrify.DirectAuthorize.Agent.Config.exe to perform those tasks.

Viewing authorization details

By default, access control, privilege management, and auditing features are enabled after you install and configure the Agent for Windows. To see details

about your rights, role definitions, role assignments, and auditing status, you can launch the Authorization Center from the command prompt.

To open the Authorization Center on a computer with the Server Core operating system:

1. Navigate to the Agent for Windows installation directory.

   By default, the agent files are installed in C:\Program Files\Centrify\Agent for Windows directory. <!---TODO update path--->

2. Run Centrify.DirectAuthorize.Auth.Center.exe. <!---TODO update filename--->

Configuring auditing options

By default, access control, privilege management, and auditing features are enabled when you install the Agent for Windows. To configure auditing options and specify the audit installation for the agent, you can launch the DirectAudit Agent Control Panel from the command prompt.

To open the DirectAudit Agent Control Panel to configure auditing features:

1. Navigate to the Agent installation directory.

   By default, the agent files are installed in the C:\Program Files\Centrify\Agent for Windows directory. <!---TODO update path--->

2. Run Centrify.Winagent.serviceconfig.exe to launch Agent <!---TODO update filename---> Configuration. Click **Add Service** to add **Auditing and Monitoring Service**. Choose an installation. Click **Setting** on the Agent Configuration for configuration.

3. Click **Configure**.

4. Select a color quality, then click **Next**.

   Because the Server Core operating system uses very few graphical elements, in most cases you should accept the default setting of Low for the color quality. This setting minimizes the storage requirements for auditing if you have enabled video capture auditing.

5. Accept the default offline data location and maximum size or type a different location, then click **Next**.

   You can also drag the slider to change the maximum percentage of the drive the offline data can consume. In most cases, however, you should leave the default setting unchanged.

6. Select the audit installation, then click **Next**.

7. Review your configuration settings, then click **Next**.

8. Click **Finish** to close the configuration wizard.

9. Click **Close** to close the control panel.

Running command line programs

The Agent for Windows includes several command line programs for performing administrative tasks. The following command line programs are supported on Server Core computers:

- dzinfo
- dzdiag
- dzrefresh
- dzflush
- dzdump
- runasrole

For more information about the command line options or output for these commands, see the *Administrator's Guide for Windows*or run the command with the /help option.

The topics in this section describe how to resolve issues with logging on, find log files, set the level of detail recorded in log files, and use diagnostic tools to retrieve information about the operation of the Agent for Windows.

Solving problems with logging on

Once you have the Agent installed on your computer, you cannot log on without a role assignment. The role, however, may be assigned to your local account, your domain account, or a remote computer. Consequently, you might encounter problems logging on after the agent is deployed. For example, you might find that you can log on to your computer using your local account but cannot log on using your domain account or have trouble connecting to a remote server.

You have no control over the roles assigned to your local, domain, or remote server accounts. These are all set by the administrator. There are a couple of things you can try if you cannot log on:

- Try to log on using a local user account or using a different domain account if you have more than one account available.
- Determine whether the computer you are using is connected or disconnected from the network. In rare cases, authorization information might not be available when a computer disconnected from the network.
- If you cannot log on to a remote computer, confirm that you have role that has the remote logon system right and that the computer is configured to allow users to log on remotely. Open the Authorization Center to see details about your roles and their rights.

Your administrator is the only person who can correct any log on problems. You should contact an administrator for your organization to proceed.

Accessing network computers with privileges

Depending on how your administrator has defined the roles you are assigned, it is possible for you to see potentially misleading information in certain applications or be unable to perform administrative tasks as you expect. For example, if you select a role with administrative privileges to access an application such as SQL Server Configuration Manager or Microsoft SQL Server Management Studio and connect to a remote SQL Server instances, it might appear as if you have permission to start and stop services or perform other tasks. However, if your role does not include network access rights for the remote SQL Server instance, you will not have the appropriate permission to perform those tasks.

You can check whether your selected role includes network access rights using the Authorization Center. If the role you are using does not include network access rights, you should click **Advanced View** to see if you have additional network roles available to use in conjunction with your local role. If the role you are using includes network access rights, you should contact your administrator to find out if those rights are applicable on the network computer you are attempting to manage.

Running diagnostics and viewing logs for the agent

The Agent for Windows provides logging and diagnostic services. If you have administrative access on a local computer, you can generate diagnostic information about the operation of the Agent for Windows and view and save the current content of the log file from the agent configuration panel. For example, you can generate diagnostic information about user sessions, user roles, desktops, and elevated account access, as well as detailed information about auditing from the agent configuration panel.

You can view these diagnostics tools either from the Windows system tray or from the agent configuration panel.

- Delinea icon in the Windows system tray - right-click it and click **Troubleshooting**, and then the service for which you want diagnostic information.
- Agent Configuration - select the service for which you want diagnostic information, then click the **Troubleshooting** tab.

Refreshing cached information

If you are a local administrator on a managed computer, you can refresh the authorization information stored in the cache to ensure the agent has the most up-to-date information about your current rights and roles. For example, if you are assigned a new role or been granted new application rights, you can refresh the cache to get the new assignment or application rights.

Checking your rights and roles using dzinfo

You can use the dzinfo command line program in a Command Prompt window to view detailed information about your rights, roles, and role assignments. The dzinfo command line utility provides the same functionality as the Authorization Center described in Checking your rights and role assignments, but allows you to view and capture the output from the command in a single window.

The syntax for the dzinfo program is:

dzinfo

The command returns detailed information about your rights, roles, and role assignment similar to the following:

```
Effective roles for AJAX\rey.garcia:
weblogic2/portland
Zone: CN=portland,CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
Status: Active

Domain Admin/portland
Zone: CN=portland,CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
Status: Active

Windows Login/mainoffice
Zone: CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
Status: Active

Effective Login Rights for AJAX\rey.garcia:
Console Login: Permitted
Audit Level: Audit if possible

Remote Login: Permitted
Audit Level: Audit if possible

PowerShell Remote Access: Permitted
Audit Level: Audit if possible

Role Assignments for AJAX\rey.garcia:
weblogic2/portland
Status: Active
Account: AJAX\rey.garcia
Scope: Zone
Zone: ajax.org/Acme/Zones/mainoffice/portland
Local Role: No
Network Role: Yes
Effective: Immediate
Expires: Never

Domain Admin/portland
Status: Active
Account: AJAX\rey.garcia
Scope: Zone
Zone: ajax.org/Acme/Zones/mainoffice/portland
Local Role: No
Network Role: Yes
Effective: Immediate
Expires: Never

Windows Login/mainoffice
Status: Active
Account: AJAX\Domain Admins
Scope: Zone
Zone: ajax.org/Acme/Zones/mainoffice
Local Role: Yes
Network Role: No
Effective: Immediate
Expires: Never

Role Definitions:
weblogic2/portland
Status: Active
Description: None
Zone: CN=portland,CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
```

Login Permitted: No
Audit Level: Audit if possible
Rescue Right: No
Require MFA: No
Available Hours:
12 2 4 6 8 10 12 2 4 6 8 10
Sunday X X X X X X X X X X X X X X X X X X X X X X X X
Monday X X X X X X X X X X X X X X X X X X X X X X X X Tuesday X X X X X X X X X X X X X X X X X X X X X X X X Wednesday X X X X X X X X X X X X X X X X X X X X X X X X
Thursday X X X X X X X X X X X X X X X X X X X X X X X X Friday X X X X X X X X X X X X X X X X X X X X X X X X Saturday X X X X X X X X X X X X X X X X X X X X X X X X
Rights:
weblogic Network Access/portland
Type: Network Access
Description: None
Priority: 0
Run As: AJAX\wladmin
Require Authentication: No

weblogic Desktop/portland
Type: Desktop
Description: None
Priority: 0
Run As: AJAX\wladmin
Require Authentication: No

Domain Admin/portland
Status: Active
Description: None
Zone: CN=portland,CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
Login Permitted: No
Audit Level: Audit if possible
Rescue Right: No
Available Hours: All
Rights:
ADUC/portland
Type: Application
Description: Active Directory Users and Computers as Admin
Priority: 0
Run As: AJAX\Administrator
Application: mmc.exe
Path: C:\Windows\system32
C:\Windows
C:\Program Files
C:\Program Files (x86)
C:\Windows\SysWOW64
Arguments: "C:\Windows\system32\dsa.msc"
Match Case: No
Require Authentication: No
Application Criteria:
None

Domain Admin Network Access/portland
Type: Network Access
Description: None
Priority: 0
Run As: AJAX\Administrator
Require Authentication: No

Windows Login/mainoffice
Status: Active

Description: Predefined system role for general Windows login users.
Zone: CN=mainoffice,CN=Zones,OU=Acme,DC=ajax,DC=org
Login Permitted: Console & Remote & PowerShell Remote
Audit Level: Audit if possible
Rescue Right: No
Available Hours: All
Rights:
None

Computer is joined to zone ajax.org/Acme/Zones/mainoffice

Auditing for AJAX\rey.garcia:
Session ID 2:
Desktops:
Default: Not currently auditing.

Auditing is not available on this computer.

Evaluations

This section contains evaluation guides for

- Windows
- Linux/Unix

Delinea

December 2021 (release 2021.1)

- Setting Up the Evaluation Environment
- How Authentication Works for Windows
- Creating and Using Roles and Desktops
- Auditing Sessions

This section describes how to prepare for an evaluation of Server Suite on a Windows computer. It includes instructions for installing Server Suite components and the Agent for Windows to enable a full evaluation of access control, privilege management, and auditing on Windows computers.

Installing and configuring Server Suite requires about a half hour. If you need to install Microsoft SQL Server Express with Advanced Features, which is also included in the package, add another 10 to 15 minutes to the setup.

Preview of Tasks

You will perform the following tasks to set up the evaluation environment. You should perform the tasks in the order shown to prepare your environment for a meaningful evaluation that demonstrates the key features of the solution for Windows computers.

1. Ensure you have at least one Active Directory *domain controller* and one Windows *domain computer*—also referred to as the Windows *client computer*.

   See Basic Requirements for the Evaluation for details about the system requirements for these computers.

2. Acquire Delinea software for the Windows client computer.

   See Downloading Delinea Software for Windows Evaluations for details about acquiring Delinea software.

3. On the Active Directory domain controller, create an Active Directory user and group to be used in the evaluation.

   See Creating an Active Directory User and Group for details about this procedure.

4. Install Access Manager and administrative tools on the Windows client computer.

   See Preparing to Evaluate Access Management for details about installing these features.

5. Use Access Manager to configure Active Directory on the domain controller.

   See Configuring Active Directory Using Access Manager for details about configuring Active Directory from Access Manager.

6. Use Access Manager to create a zone.

   See Creating the First Zone for details about creating a zone.

7. Use Access Manager to assign the Windows Login role to your Active Directory account.

   See Assigning Yourself the Default Windows Login Role for details about this procedure.

8. If you are evaluating auditing features, you need access to an instance of Microsoft SQL Server and an audit installation, which consists of several auditing-specific components.

   See Identifying a Microsoft SQL Server Instance for details about installing a SQL Server Express instance for demonstration purposes.

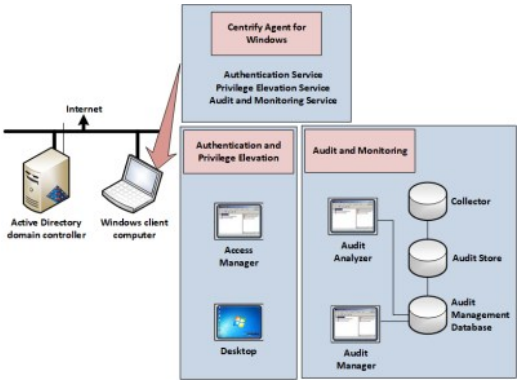
   See Preparing to Evaluate Auditing for details about installing audit components.

9. Install the Agent for Windows on the Windows client computer.

   See Installing the Agent for Windows for details about this procedure.

Basic Requirements for the Evaluation

The installation procedures described in this guide are based upon a minimal configuration with one Windows Delinea-managed computer (the *Windows client computer*) and one Windows Active Directory domain controller, as illustrated in the following figure.



You can add more Windows computers to the configuration to expand the scenario or to make the evaluation more consistent with a production deployment.

Preparing an Active Directory Domain Controller

You must have an Active Directory domain controller to use in this evaluation. You should also have a properly configured Domain Name Service that enables the computers used in the evaluation to communicate.

For details about supported platforms, please consult the release notes.

> **Note**: In the configuration illustrated in the previous topic, no software is installed on the domain controller. However, because this is just an evaluation environment, you could install all of the software on the domain controller. In a production environment it is not recommended to install this software on domain controllers.

Selecting a Windows Domain Computer

The Windows client computer that you use for the evaluation should have a supported Windows operating system and minimum system requirements.

Windows operating system	Windows 7 or later or a Windows server platform. Please consult the release notes for supported platform versions.
.NET Framework	.NET Framework 4.6.2 or later If .NET is not installed, the setup program will install it for you.
CPU speed	Minimum 2 GHZ
RAM	4 GB
Disk space	20 GB free space

Desktop rights can be used on Windows servers and workstations that have a traditional Windows desktop. If the computer you are using is running Windows Server 2012 or 2012 R2, Windows does not provide access to applications natively when you switch from the default desktop to a privileged desktop due to changes to the underlying interfaces and supported features within the operating system. To enable access to applications on computers running these versions of Windows, the Agent for Windows provides a custom start menu. The Delinea start menu allows you to open and run applications as you would on Windows 7 or Windows Server 2008 R2. The Delinea start menu is installed on the left side of the taskbar and displays the Delinea logo. This start menu is only available if you are using a role with Delinea desktop rights and cannot be modified.

Delinea also recommends that you install the Microsoft Windows Server Administration Tools Pack on the computer on where you install Access Manager. The Administration Tools Pack includes the Active Directory Users and Computers utility—`dsa.msc`—used in many of the exercises.

If you are using the recommended configuration with a separate Windows client computer that is not the domain controller, be sure that the Windows client computer is joined to the Active Directory domain.

Downloading Delinea Software for Windows Evaluations

You can go to the this website to sign up for a free trial. Once you're signed up with an account, you can download the software.

To register for a free trial:

1. Navigate to https://www.delinea.com/free-trial/.
2. Enter your contact and company information, click the checkbox to indicate that you agree to the terms of use and privacy policy, and then click **Start My Trial**.

   Note: You will receive an email with the next steps in downloading your free trial.

Downloading Server Suite Software for Windows Evaluations

You can download all of the components for Server Suite from the Delinea website to your Windows computer. Before you begin, be sure you have the email address and password you used to register for your trial.

To download the Windows software for Server Suite:

1. Open a browser on the Windows computer you plan to use for the evaluation and go to https://www.delinea.com.
2. In the upper area of the web page, click **Login**.
3. Enter your email address and your account password, then click **Login**.
4. Go to **Support** > **Downloads**.
5. Select **Zero Trust Privileges - Enterprise** to locate the latest software bundles.
6. Next to the latest version for 64-bit Windows systems, click either the **ISO** or **ZIP** button to download the software in that format.

   The latest version of the Windows software bundle is called Server Suite.
7. Close the window when the download is complete.

Creating an Active Directory User and Group

Evaluation scenarios covered in this guide require an Active Directory user with normal user privileges to demonstrate different features. For example, you will create access rights that grant elevated privileges to a role and assign this user to the role to use those rights.

To prepare for the evaluation scenarios:

1. On the Active Directory domain controller, open Active Directory Users and Computers.

   For example, create the user amy.adams to represent a domain user with a valid logon account.
2. Select **Action > New > User** and follow the prompts to create a new Active Directory user.
3. Select **Action > New > Group** and follow the prompts to create a new Active Directory group.

   For example, create the group Eval Group to represent a typical Active Directory security group to which you would assign a role.
4. Right-click the user name and select **Add to a group** to add the new user to the new group.

   You might also want to add your own Windows account to the new group. Adding your own account to the Evaluation group makes it easier to demonstrate some features, such as assigning roles to group members.

Preparing to Evaluate Access Management

If you are evaluating access control and privilege management features, you must install the administrative tools on the Windows client computer to prepare for the evaluation. Later, you will also install the Agent for Windows on the Windows client computer as described in Installing the Agent for.

To install Access Manager from the installer:

1. Log on to the Windows client computer using a Windows account that has Active Directory administrator privileges on the domain controller.
2. From the Delinea CD or directory that has Delinea software, open **autorun**.

3. On the Getting Started page, click **Authentication & Privilege** to start the setup program for Authentication Service and Privilege Elevation Service.
4. Follow the prompts displayed and select Administration as the components to install.

   For a Windows-only evaluation, none of the Utilities components are applicable.
5. Accept the defaults for the remaining selections, then click **Finish** to close the setup program.

Configuring Active Directory Using Access Manager

The setup program adds shortcuts for selected components to your desktop to give you immediate access to the consoles you will use. Before you can use Access Manager to create zones, define access rights and roles, and assign roles to users and groups, however, you use it to run a Setup Wizard that prepares the Active Directory forest with parent containers for licenses and zones.

To use the Setup Wizard to configure Active Directory:

1. From the desktop, open Access Manager.
2. Select **Use currently connected user credentials** to use your current log on account, then click **Next**.
3. Select **Generate recommended deployment structure** and **Generate default deployment structure**, then click **Next**.
4. Click Browse to select the container you would like to use for the deployment structure.

   You can select any domain in the forest, including the forest root domain.
5. Select a location for installing license keys in Active Directory, then click **Next**.

   The Setup Wizard displays information about the Read permissions that must be granted on the container. Click **Yes** to continue.
6. Type, copy and paste, or import the license key you received, click **Add**, then click **Next**.
7. Click **Next** to use the default container for zones.
8. Click **Next** to skip the following options:
   - Grant computer accounts permission to update their own account information.
   - Register the administrative notification handler.
   - Activation of profile property pages.
9. Review the summary, click **Next**, then click **Finish**.

The wizard opens the Access Manager console. For reference, the user account under which you are logged in displays in the main panel just below **Access Manager**.

Creating the First Zone

In this section, you create a zone for the Windows client computer. After you create the zone, you can start creating access rights, defining roles, and assigning roles to Active Directory users and groups.

To create a new zone:

1. In Access Manager, click **Create Zone**.

2. Type a name and description for the zone, for example Headquarters, then click **Next** to accept the defaults for the other fields.
3. Click **Finish**.

You now have one parent zone in Access Manager. Expand **Access Manager > Zones** to view your new zone in the console.

Assigning Yourself the Default Windows Login Role

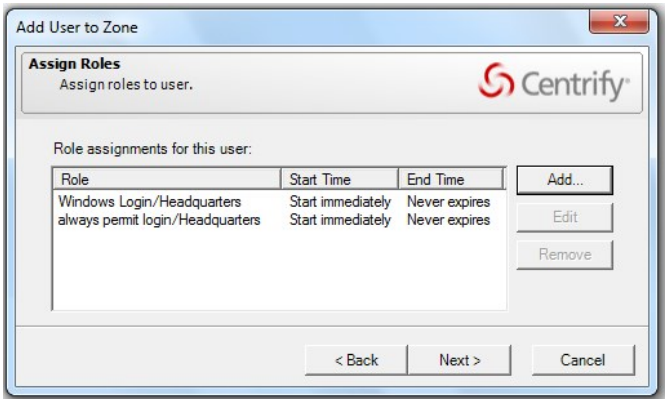
After you install the Agent for Windows, you must be assigned to a role that allows you to log on. To finish the preparation of the evaluation environment for access control and privilege management, you are going to assign a role with the log in privilege to your Active Directory account. The Windows Login role is a predefined role that grants permission to log on locally and connect remotely for Delinea-managed Windows computers.

To assign the Windows Login role to your account:

1. In Access Manager, expand Zones and select the zone you created in Creating the First Zone.
2. Right-click the zone, and select **Add User**.
3. Select **Active Directory user** and click **Next**.
4. Type the path to your account or click **Browse** to search for and select your Active Directory user account, then click **Next**.

   For example, click Browse and type all or part of the name, then click **Find Now**. You can then select your account name in the list of results and click **OK**.
5. Deselect **Define user UNIX profile** and make sure **Assign roles** is selected, then click **Next**.
6. Click **Add**, select the predefined **Windows Login** role, and click **OK**.
7. Check the role assignment start and end times for your account are set to Start immediately and Never expire, then click **OK**.
8. Repeat Step 6 and Step 7 to add the **Rescue - always permit login** role.

   Your Add User to Zone window should show the following roles:



9. Click **Next**, then click **Finish**.

If you are evaluating auditing features, go on to Preparing to Evaluate Auditing. If you are only evaluating access-related features, skip to Installing the Agent for Windows.

Preparing to Evaluate Auditing

If you are evaluating access and auditing features or only auditing, there are several components that make up the auditing infrastructure. For evaluation, you can install all of the components on the same computer.

Identifying a Microsoft SQL Server Instance

If you are evaluating both access and auditing features or only auditing, you must have at least one Microsoft SQL Server instance for storing audit-related information.

For evaluation purposes, you can use an existing Microsoft SQL Server database instance to which you have administrative access or automatically install and configure an instance of Microsoft SQL Server Express with Advanced features directly from the Audit Configuration Wizard.

You should only use Microsoft SQL Server Express for evaluation and testing. You should not use Microsoft SQL Server Express for a production environment.

Installing the Auditing Components

In this section, you run the setup program to install the auditing and monitoring components, including the Audit Manager and Audit Analyzer consoles, on the Windows client computer.

To install the Audit & Monitoring Service from the installer:

1. Log on to the Windows client computer using a domain account with administrative privileges, such as DEMO\\administrator. Do not log on as a local user.
2. From the Delinea CD or directory that has Delinea software, open **autorun**.
3. On the Getting Started page, click **Audit & Monitor** to start the setup program for the audit and monitoring service.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
6. Select **Administration** and **Services** to install both Audit Manager and Audit Analyzer, then click **Next**.
7. Accept the defaults for the remaining selections and confirm that the **Launch Configuration Wizard** option is selected, then click **Finish** to close the setup program.

   Note If the **Launch Audit Configuration Wizard** option is not selectable, a possible cause is that you are logged on to the Windows client computer as a local user (for example, local administrator) rather than as a user with domain administrative privileges. In this scenario, select **Start > Switch user** and log on as a Windows domain user with administrative privileges (for example, DEMO\\administrator). Then launch Audit Manager from the desktop icon, and select **Action > New Installation** to start the audit configuration wizard.

8. In the Welcome page for the audit configuration wizard, click **Next**.
9. Select **Create a new installation** and type a name for your installation, then click **Next** to capture audit trail events without recording video of an audited user's desktop activity.

   Audit trail events provide a summary of user activity, for example, when users log on and off, open and close applications, and use role assignments with elevated rights. If you want to be able to review what was displayed on the screen during an audited user's session, you can select **Enable video capture auditing of user activity**. This option increases the database storage required for auditing.
10. Select **Install a new SQL Server Express instance on this computer** and specify the instance name, then click **Next**.
11. Verify the default path to the Microsoft SQL Server Express setup program, the disk space requirements, and the location for the files, then click **Next**.

    Note: If an incompatible instance of SQL Server Express is already installed, the wizard displays an error message instructing you to uninstall that instance. Use the Windows control panel to uninstall the incompatible instance of SQL Server Express, and then try the SQL Server Express installation from the wizard again.
12. Review the summary, then click **Finish**.

The audit configuration wizard automatically configures the audit store scope, audit store database, and a collector on the local computer. After the auditing infrastructure is in place, you can install the Agent for Windows and join the computer to the zone you created.

Installing the Agent for Windows

You are now ready to install the Agent for Windows on the client computer to begin the evaluation. In a production environment, you would install the agent on all of the Windows computers in the domain that you want to manage or audit.

> **Note**: The following instructions assume you are still logged in with your administrator account. Be sure that this account has at least the Windows Login and Rescue - always permit login roles assigned as described in Assigning Yourself the Default Windows Login Role to ensure you can log on after the agent is installed. If the account you are using to install the agent does not have the Windows Login role assigned, the agent configuration wizard will allow you to assign the Windows Login role to the domain administrators (Domain Admins) group when you join a zone.

To install the Agent for Windows using the setup program:

1. Insert the Delinea distribution CD into the computer on which you wish to install the agent or browse to the location where you have saved downloaded Delinea files.
2. On the Getting Started page, click **Agent** to start the setup program for the agent.

   If the Getting Started page is not displayed, open the autorun.exe file to start the software installation.
3. If a previous version of the agent is installed, click **Yes** when prompted to upgrade the Agent for Windows.
4. At the Welcome page, click **Next**.
5. Review the terms of the license agreement, click **I accept the terms in the License Agreement**, then click **Next**.
6. Accept the default location for installing components, or click **Change** to select a different location, then click **Next**.
7. In the Ready to install Agent for Windows page, click **Install**.
8. Click **Finish** to complete the installation and start the agent configuration panel.

You must restart the computer after you configure the Privilege Elevation Service. When prompted, click **Yes** to restart the computer immediately.

After you restart the computer, log on with your administrator account. Left-click on the Delinea icon on your taskbar to confirm that you are viewing your default desktop. In the next chapter, you will see how to configure access rights and roles and how to select from roles you are assigned.

Configuring the Agent

By default, when you click **Finish**, the setup program opens the agent configuration panel. In the agent configuration panel, you can enable the agent to connect to Delinea services that are installed on the main administrative computer as described in Installing the Agent for Windows. After a service is enabled, you can use the agent configuration panel to configure settings that define how the agent will interact with each service.

The first time the agent configuration panel opens, it does not display any services for you to enable. Services display in the agent configuration panel only after you manually instruct the configuration panel to check for services and display those that are eligible to be enabled.

Only services that are installed and configured as required are eligible to be enabled. For example, if you installed the Privilege Elevation Service earlier (as described in Preparing to Evaluate Access Management) but did not create a zone, the Privilege Elevation Service does not display on the list of services that you can enable.

To enable services using the agent configuration panel:

1. If the agent configuration panel is not open, open it by clicking **Agent Configuration** in the list of applications in the Windows Start menu.
2. In the agent configuration control panel, click **Add service**.

   All services that are available to be enabled are displayed.
3. In the list of services, highlight a service and click **OK**.
4. Provide additional information about the service that you are enabling:
   - **Audit & Monitoring Service**:

     In the Select an Audit Installation page, select an audit store from the list of available audit stores. Click **Next**, and the computer is connected to the audit store.
   - **Identity Platform Settings**:

     1. In the Connect to Identity Platform page, type the URL of the identity platform instance to connect to, or select an instance from the list of registered platform instances in the forest. Click **Next**.
     2. In the Multi-factor authentication for Windows Login page, ensure that the check box to enable multi-factor authentication is selected. Next, use the **All Active Directory accounts** button or **Accounts below** button to specify which Active Directory accounts are enabled for multi-factor authentication login. If you select **Account below**, use the **Add** and **Remove** buttons to select accounts. Click **Next** when you are finished.
   - **Privilege Elevation Service**:

     1. In the Join to a zone page, type a zone or select a zone from the list of available zones. You can also choose to select the option to retrieve the zone data before the computer restarts. This option can be helpful in situations where you might lose connection to the domain after restarting, such as when you're using a VPN connection.

        Click **Next**, and the computer is joined to the zone.
     2. After the computer is joined to a zone, you must reboot the computer to activate all privilege elevation service features on the computer.

        If the zone that you select is already configured with a Privileged Access Service tenant, the message **Identity Platform enabled** displays after the computer joins the zone. In this situation, the instance is managed by the zone, and is shown as read-only.
5. To add additional services, click **Add service** and repeat the preceding steps.

   When you are done, the services that you enabled are shown in the **Enabled services** section of the agent configuration panel.
6. If necessary, continue to configure services after their initial configuration during enablement as described in these sections:
   - Configuring Agent Settings for the Audit and Monitoring Service
   - Configuring Agent Settings for Offline Audit and Monitoring Service Storage
   - Configuring Agent Settings for the Identity Platform
   - Configuring Agent Settings for Privilege Elevation

Configuring Agent Settings for Audit and Monitoring Service

If you want to reconfigure agent settings for auditing on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the service), you can open the agent configuration panel manually and configure the agent as described in this section.

To configure agent settings for audit and monitoring service:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

   The agent configuration panel opens, and displays the services that are currently enabled. You can configure any service listed in the **Enabled services** section.

2. Click **Audit & Monitoring Service**, and then click **Settings**.

3. In the General tab, click **Configure**.

4. Select the maximum color quality for recorded sessions, then click **Next**.

   See Selecting the Maximum Color Quality for Recorded Sessions for more information on the configuration of this setting.

5. Specify the offline data location and the maximum percentage of disk that the offline data file should be allowed to occupy, then click **Next**.

   See Configuring Agent Settings for Offline Audit and Monitoring Service Storage for more information on the configuration of this setting.

6. Select the installation that the agent belongs to, then click **Next**.

7. Review your settings, then click **Next**.

8. Click **Finish**.

9. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see the *Administrator's Guide for Windows*.

Selecting the Maximum Color Quality for Recorded Sessions

Because auditing Windows computers captures user activity as video, you can configure the color depth of the sessions to control the size of data that must be transferred over the network and stored in the database. A higher color depth increases the CPU overhead on audited computers but improves resolution when the session is played back. A lower color depth decreases network traffic and database storage requirements, but reduces the resolution of recorded sessions.

The default color quality is low (8-bit).

Configuring Agent Settings for Offline Audit and Monitoring Service Storage

The "Maximum size of the offline data file" setting defines the minimum percentage of disk space that should be available, if needed, for audit and monitoring service. It is intended to prevent audited computers from running out of disk space if the agent is sending data to its offline data storage location because no collectors are available.

For example, if you set the threshold to 10%, auditing will continue while spooling data to the offline file location as long as there is a least 10% of available disk space on the spool partition. When the available disk space reaches the threshold, auditing will stop until a collector is available.

The agent checks the spool disk space by periodically running a background process. By default, the background process runs every 15 seconds. Because of the delay between background checks, it is possible for the actual disk space available to fall below the threshold setting. If this were to occur, auditing would stop at the next interval. You can configure the interval for the background process to run by editing the HKLM\Software\Centrify\DirectAudit\Agent\DiskCheckInterval registry setting.

Configuring Agent Settings for the Identity Platform

If you want to reconfigure agent settings for Privileged Access Service on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the service), you can open the agent configuration panel manually and configure the agent as described in this section.

To configure agent settings for the Identity Platform:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

   The agent configuration panel opens, and displays the services that are currently enabled. You can configure any service listed in the **Enabled services** section.

2. Click **Identity Platform**, and then click **Settings**.
3. In the General tab, review the Status field in the Features area:
    - If the status is **Enabled**, the computer is not joined to a zone, and you can configure all Identity Platform settings that are shown in the General tab.
    - If the status is **Enabled per zone settings**, the computer is joined to a zone, and most Identity Platform settings are based on the zone configuration. In this situation, the **Browse** and **Details** buttons in the General tab are disabled, because those features are controlled by the zone configuration. The only configuration that you can perform in the General tab is to change the proxy server settings.
4. To change proxy server settings:
    1. Click **Change**.
    2. Specify a new proxy server address.
    3. Click **OK**.
5. To change to a different Identity Platform instance (only configurable if the computer is not joined to a zone):
    1. Click **Browse**.
    2. Select an instance from the list of registered platform instances in the forest.
    3. Click **OK**.
6. To specify which Active Directory accounts require multi-factor authentication (only configurable if the computer is not joined to a zone):
    1. Click Details.
    2. Use the **All Active Directory accounts** button or **Accounts below** button to specify which Active Directory accounts are enabled for multi-factor authentication login. If you select **Account below**, use the **Add** and **Remove** buttons to select accounts.
    3. Click **OK**.
7. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see the *Multi-factor Authentication Quick Start Guide*.

Configuring Agent Settings for Privilege Elevation

If you want to reconfigure agent settings for privilege elevation on a Windows computer after initially configuring them during enablement (or if you did not use the agent configuration panel when you enabled the privilege elevation service), you can open the agent configuration panel manually and configure the agent as described in this section.

To configure agent settings for privilege elevation:

1. In the Windows Start menu, click **Agent Configuration** in the list of applications.

    The agent configuration control panel opens, and displays the services that are currently enabled. You can configure any service listed in the **Enabled services** section.
2. Click **Privilege Elevation Service**, and then click **Settings**.
3. In the General tab, click **Change**.
4. In **Change the zone for this computer**, click **Browse**.
5. Click **Find Now** to search for an appropriate zone for the agent.
6. Select a zone from the list of search results, then click **OK**.
7. Click **OK** to use the zone you selected.
8. Click **Close** in the General tab to save your changes.

For information about using the Troubleshooting tab, see the *Administrator's Guide for Windows*.

This chapter introduces core concepts and features that you should be familiar with before starting an evaluation of Delinea software for managing Windows computers.

Providing Access Control and Accountability

In many organizations, most computer users are given very restricted access privileges to minimize the exposure of sensitive services and data to possible compromise. However, there are often a few applications, procedures, or services that require enhanced privileges and to which these users need access. For example, a user might occasionally have to install software or run a restricted internal application. For this purpose, these organizations often provide these users with login information for accounts with enhanced privileges. Unfortunately, this policy substantially undermines security, because there's no way to tell—even on an audited system—who actually logged on to these accounts, and once logged on, a malicious user is not restricted to the procedures for which he was given the login information in the first place.

Delinea solves this problem by enabling you to assign roles that give a user access to only those services or applications and restricted access privileges only when the user needs them.

For Windows computers, Delinea provides three main services: access control, privilege management, and auditing. These services can be used together or independently.

To provide access control, privilege management, and auditing for Windows computers, Delinea relies on the following:

- **Authentication Service** and **Privilege Elevation Service** features enable you to define access control privileges, create roles composed of a set of privileges, and assign users or groups to those roles. You can also use zone technology to limit the scope of a role to limited sets of computers. You can, also, configure roles with start and expiration dates or to be active on specific days of the week and hours of the day.
- **Audit & Monitoring Service** enables you to collect and store an audit trail of user activity and provides a console for searching and replaying captured sessions.
- **Agent for Windows** enables you to deploy access and auditing features on the Windows computers you want to manage.

You can use Privilege Elevation Service without auditing if you aren't interested in collecting and storing information about session activities. You can also deploy Server Suite without access and privilege management features if you are only interested in auditing activity on Windows computers. However, the real value of Delinea software for Windows computers comes from using the services together as an integrated solution for managing elevated privileges and ensuring regulatory compliance across all platforms in your organization. That way you can restrict access to only those instances when elevated permissions are absolutely necessary, and audit only user activity that merits auditing.

Organizing Computers and Access Rights

This guide is intended to help you evaluate how you can use Delinea software to manage access and administrative privileges for Windows computers and applications. However, Delinea also enables you to include UNIX, Linux, and Mac OS X computers in Active Directory, providing you with a single repository for all managed computers, users, privileges, and roles. Delinea enables this cross-platform integration through the use of **Zones**.

A zone is a logical object that you create using Access Manager. You use the zone to organize computers, rights, and roles into groups. In each group, you can define different access rights, different role availability rules, and different role assignments. You can create the zones in a hierarchy of parent and child zones, so that rights and roles can inherited or zone-specific.

As part of the evaluation, you will create a zone for the Windows computers, define access rights that are specifically for Windows computers, create roles that include those access rights, and assign roles to users and groups.

Restricting Access to Administrative Privileges

By defining roles with specific access permissions, you can use Access Manager to specify the conditions under which users can perform privileged operations. A user logs on to the Windows computer with his or her normal, restricted login, and then selects the role they need to perform a privileged operation only when that access is needed. You can restrict a role or desktop to certain times or days of the week, and you can set a beginning and expiration date for the access. You can set any role or desktop to require auditing, so that the user cannot use the role or desktop unless it is being audited.

Access Manager provides three kinds of Windows access rights. For Windows computers, these specialized access rights are:

- **Desktop** access rights enable you to create additional working environments and run any application in that desktop as a member of Active Directory or built-in group.
- **Application** access rights enable you to run a specific local application as another user or as a member of an Active Directory or built-in group. This access right is similar to the standard **Run as** menu option, except that someone assigned a role with this right doesn't need to know the privileged

user's password to use it.
- **Network** access rights enable you to connect to a remote computer as another user or as a member of an Active Directory or built-in group to perform operations, such as start and stop services, that require administrative privileges on the remote computer.

You configure these access rights using the Access Manager console. The rights are enforced through an Agent for Windows installed on each computer you want to manage.

Auditing User Activity on a Managed Computer

When you install the Agent for Windows on a computer, you have the option to enable access management, auditing, or both. If you enable auditing features, the agent can capture detailed information about user activity and all of the events that occurred in each user session on the managed computer. The user activity captured includes an audit trail of the actions a user has taken and a video record of everything displayed on the screen. For users who have privileged access to computers and applications, the audit and monitoring service helps ensure accountability and improve regulatory compliance. By recording user sessions, you can see exactly who had access to which computers and what they did, including any changes they made to key files or configurations.

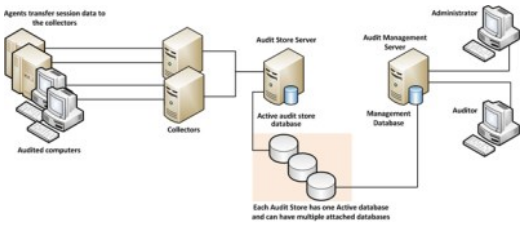
The audit and monitoring service collects user activity as it occurs. The recorded activity is transferred to a Microsoft SQL Server database so that it is available for querying and playback. You can search the stored user sessions to look for policy violations, user errors, or malicious activity.

To ensure scalability and enterprise readiness, the auditing infrastructure consists of multiple components called a **audit and monitoring service installation**:

- **Audited computers** are the computers on which you want to monitor activity. To be audited, the computer must have the Agent for Windows installed with auditing enabled and be joined to an Active Directory domain.
- One or more **collectors** receive the captured activity from the agents on audited computers and forward it to an audit store database.
- An **audit store** defines a scope, such as an Active Directory site or a subnet, and one or more databases that store captured activity and audit trail records from the collectors and store it for querying.
- A **management database** keeps track of all the agents, collectors, and audit stores that make up a single DirectAudit installation.
- **Consoles** enable administrators to configure and manage all of the audit-related components and auditors to query and review user sessions.

When you enable auditing on a computer with the Agent for Windows, the agent captures user activity on that computer and forwards it to a collector computer. If no collectors are available, the agent caches the session data locally and transfers it to a collector later. The collector sends the data to an audit store database. When administrators or auditors want to review the captured data, they use the Audit Analyzer to search for and play back the session. The Audit Analyzer connects to the management database which retrieves the data from the appropriate audit store. The administrator can control the audit data available to any specific user or group through auditor roles that limit audit access rights and privileges.

The following figure illustrates the basic architecture and workflow in a small scale installation.



At the end of the last chapter, you restarted the Windows client computer and logged in with your administrator account.

This chapter describes how to define access rights and create roles that grant elevated privileges, assign roles to users and groups, and view details about the rights and roles available. This chapter also shows you how to select and switch between roles for running local applications and connecting to network computers.

Verifying that your Account is Assigned Basic Login Rights

At this point, you should be logged on to the Windows client computer with an administrator account that has been assigned the "Windows Login" and "Rescue always permit login role" predefined roles as described in Assigning Yourself the Default Windows Login Role. Because the client computer has Access Manager and the Agent for Windows installed, you can verify that your account has been assigned these predefined roles using Access Manager or the Authorization Center.

To use Access Manager to verify your assigned roles:

1. Expand Zones and the zone you created in Creating the First Zone.
2. Expand Authorization and select Role Assignments.

   In the right pane, you should see your role assignments displayed similar to this:

In this example, the Windows Login and Rescue roles are assigned to the Administrator account in the zone named Headquarters

Assigning the Windows Login Role to a Group

After a computer joins a zone, users must be granted access to that computer by being assigned a role with the right to log on. So far, only the Administrator account has that privilege. This exercise illustrates how you can give that privilege to other users through their Active Directory group membership.

In most cases, you can assign the Windows Login role to all local Windows users, all Active Directory users, or both, if you want to automatically allow new users to log on locally or remotely. However, the Window Login role does not override any native Windows security policies. For example, if the Local Security Policy on the domain controller does not allow Domain Users to log on locally, assigning the Windows Login role to the Domain Users security group will not allow members of that group to log on locally.

If the Windows client computer you are using for the evaluation does not allow users to log on locally or does not accept remote desktop connections, you might have to make Eval Group a member of a specific Windows security group, such as Server Operators or Remote Desktop Users, to complete further exercises.

To assign the Windows Login role to an Active Directory group:

1. On the Window client computer, open Access Manager.
2. Expand the zone, then expand Authorization.
3. Right-click **Role Assignments** and select **Assign Role**.
4. Select **Windows Login** from the list of role definitions, then click **OK** to display Assign Role.

   By default the role is set to start immediately and never expire.
5. Select **Accounts below** to assign the role to the group you created in Creating an Active Directory User and Group.

   For purposes outside of this exercise, you could assign the role to more users by selecting **All accounts** and then specifying **All Active Directory** accounts, **_All local Windows accounts**, **All local UNIX accounts**, or any combination of these three selections.
6. Click **Add AD Account** to display Add User Role Assignment.
7. Change the **Find** filter from **User** to **Group**.

8. Type all or part of the group name, click **Find Now**, then select the group in the results and click **OK**.

   For example, type Eval to search for Eval Group and select that group in the results.

9. Click **OK** to complete the assignment and close the Assign Role window.

   Now all members of Eval Group can log on to this computer.

To verify the role assignment, you can log off as the administrator and log in as the user you created in Creating an Active Directory User and Group, for example, amy.adams. When you log on using the new account, the default desktop has no administrative privileges. For example, the new user cannot stop or start services on the local computers because the account do not have the administrative privileges required to do so. The next exercise shows you how to give a user elevated privileges when she is running a specific application.

Adding Predefined Rights to a Zone

There are many predefined rights available that grant access to specific Windows applications. For example, there is a predefined Performance Monitor right that allows you to run Performance Monitor on a computer without being a local administrator or knowing an administrative password.

You can add any or all of these predefined rights to any zone so they are available to include in role definitions. Alternatively, you can add predefined rights to individual role definitions without adding them to zones. In either case, you create grant predefined rights in the context of a role definition.

To add predefined rights to a zone and the Windows Login role:

1. On the Windows client computer, open the Access Manager console.
2. Expand **Zones** and the parent zone or child zones until you see the zone (for example, Headquarters) where you want to add predefined rights.
3. Expand **Authorization > Role Definitions**.
4. Select the Windows Login role definition, right-click, then select **Add Right**.
5. Select **Any Windows Rights** from the Type list to filter the list of rights displayed.
6. Select the Headquarters zone from the list of zones, and then click **Create Predefined Rights**.

   The list of predefined rights that you can add to the Headquarters zone and to the Windows Login role is displayed. In the next steps, you will select which rights to add to the Headquarters zone. From the rights that you add to the Headquarters zone, you will select which, if any, to also add to the Windows Login role.

7. From the list of predefined rights, select the rights that you want to add to the Headquarters zone and to the Windows Login role, and then click **OK**.

   By default, all of the predefined rights that you select will be added to the Headquarters zone and to the Windows Login role. In the next step, you will deselect rights so that they are added only to the zone and not to the role.

8. Deselect predefined rights that you do not want to add to the Windows Login role.

   Rights that you deselect are added only to the Headquarters zone. Rights that you leave selected are added to both the Headquarters zone and the Windows Login role.

9. Click **OK** to add the predefined rights to the zone, role, or both according to your selections in Step 8.

   If you deselected all available predefined rights, the **OK** button is not available to click. In this scenario, click **Cancel** to add the rights to the zone without adding them to the role definition.

   After you perform this step, the predefined rights that you deselected are not added to the Windows Login role, but are added to the Headquarters zone so that they can be added later to roles in the zone as needed.

You can click **Refresh** in Access Manager to see the predefined rights listed as Windows application rights.

Creating an Application Right

An application right lets you run a specific application as a different user. An administrator assigns an application right rather than a desktop right when the user needs only occasional administrative responsibilities for a specific application and needs only temporary or infrequent use of the elevated privileges.

(Desktop rights provide administrative access to more than a single application at a time. See Creating a Desktop Right for details about desktop rights.)

If you have completed the exercises in the previous sections, you are ready to create your first application right. If you have not completed all of the exercises to this point, you might not be able to perform all of the following exercises successfully.

In the following exercises, you will:

- Verify the Active Directory domain user amy.adams does not have permission to use the Windows Control Panel to change security settings.
- Configure a new application right that gives administrative privileges for the Control Panel application.
- Define a new role that uses the application right.
- Assign the role definition that includes the Control Panel application right to the Active Directory domain user amy.adams.
- Verify that the role assignment grants the user amy.adams the right to change a setting in Control Panel.

To verify the user does not have administrative privileges for the Control Panel:

1. On the Windows client computer, log on as the amy.adams domain user account.
2. Use Windows Explorer to open the C:\Windows\System32 folder.
3. Create a shortcut for the control.exe program on the desktop.
4. Use the shortcut to open the Control Panel, select System and Security, then open **Allow a program through Windows Firewall**.

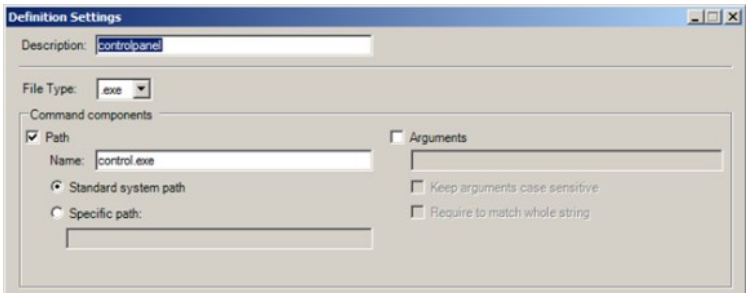
   Notice that you cannot make changes to the list of Allowed programs and features. If you click Change Settings, you are prompted to enter an administrator account name and password.
5. Click **Cancel** to close **Allow programs to communicate through Windows Firewall** and close the Control Panel.
6. Log off as amy.adams and log on with your administrator account.

To create a new application right for the Control Panel:

1. On the Windows client computer, open Access Manager and expand to display **Authorization > Windows Right Definitions**.
2. Select **Applications**, right-click, then select **New Windows Application**.
3. On the General tab, type Control Panel Right for the name of this application right and an optional description.
4. Click the Match Criteria tab, then click **Add**.

   In the Match Criteria tab, you specify one or more application executable files to be included in this application right. You can specify application executable files in many ways. The capability to specify more than one executable file in a single application right takes into account situations in which one application might reside in different locations on different computers. For details about different ways of specifying executable files, see the "Defining desktop application rights" help topic in the Access Manager online help.

   In this example, you will specify one application executable file using the application executable name and path.
5. Type a name for the criteria definition, select **Path**, then type the application executable name control.exe to specify the Windows Control Panel as the application to which this right grants access. For example:



6. Click **OK** to use the default standard system path for the application without specifying any other criteria.
7. Click the **Run As** tab, select **Self with added group privileges**, then click **Add Built-in Groups** to select the administrative group to use.

For the evaluation, you should use a built-in group to avoid adding test users and groups to your Active Directory environment. Alternatively, you could specify an existing user account, create a new user account for this right, or select **Self with added group privileges**, then click **Add AD Groups** to search for and select a previously-defined Active Directory group with administrative privileges.

8. Select the **Administrators** group, then click **OK**.
9. Select **Re-authenticate current user** to require users to authenticate their identity when they use a role with this right.
10. Select **Require multi-factor authentication** If you would like to enable multi-factor authentication for the right.

    Before you enable multi-factor authentication, you should be aware that multi-factor authentication for Delinea-managed Windows computers relies on the infrastructure provided by Privileged Access Service. For more information on preparing to use multi-factor authentication, see the *Multi-factor Authentication Quick Start Guide*.

11. Click **OK** again to complete the definition of the application right.

    The new application right is now defined. Next you must create a new role definition to use the application right.

12. To update the list of application rights in Access Manager so that you can review the new application right, select **Action > Refresh**.

To define a new role with an application right for the Control Panel:

1. Select **Role Definitions**, right-click, then select **Add Role**.
2. In the General tab, type ControlPanelAdmin as the name of the new role.

   Do not change the default settings for the System Rights tab and the Audit tab.

3. Click **OK**.

   The new role definition is created, but the role does not have any rights yet.

4. Select the ControlPanelAdmin role listed under **Role Definitions**, right-click, then select **Add Right**.
5. Select **Control Panel Right** in the list of rights, then click **OK**.

   You can filter the list of rights. For example, you can filter rights by name, type, zone, or description. After you select the right and click OK, the role definition has one right. You can add other rights to it. After you have identified all of the access rights for the role definition, you can assign the role to a user or group.

To assign the role definition with the application right to a user or group:

1. Select **Role Assignments**, right-click, then select **Assign Role**.
2. Select ControlPanelAdmin in the list of role definitions, then click **OK** to display **Assign Role**.
3. Click **Add AD Account** to search for and select the user amy.adams, then click **OK**.
4. Select **Role Assignments** to see that the user amy.adams is assigned the Windows Login and ControlPanelAdmin roles.
5. Open the **Privilege Elevation Service Settings** (from the Agent Configuration shortcut > Privilege Elevation Service > Settings), click the **Troubleshooting** tab, then click **Refresh** to force the agent to get the latest authorization information without waiting for the cache to expire.

To verify the user has administrative privileges for the Control Panel:

1. Log off as the administrator and log in as amy.adams.
2. Right-click the control.exe shortcut on the desktop.

   If you want to open an application from the Start menu, press the Shift key when you right-click.

3. Select **Run with Privilege**.

   Selecting **Run with Privilege** is similar to selecting standard Windows “Run as” or “Run as administrator” menu items, but does not require you to provide a password for an administrative or shared service account. Instead, you always use your own password to authenticate your identity.

4. Select ControlPanelAdmin in the list of the roles available, then click **OK**.
5. Type the password for the amy.adams login account, then click **OK**.

6. Select System and Security, then open **Allow a program through Windows Firewall**.

   Notice that you can now make changes to the list of programs allowed through the firewall.

This section showed you how to set up a role that allows privilege escalation for a single application and how the user can select that role to run the application with privileges without knowing the administrator's user name or password.

Creating a Desktop Right

In the preceding section, you saw how to elevate privileges by creating an application right for a specific application. To grant administrative access to more than a single application at a time, you can allow users to open a desktop that has administrative privileges.

If you have completed the exercises in the previous sections, you are ready to create your first desktop right. If you have not completed all of the exercises to this point, you might not be able to perform the following exercise successfully.

In the next exercise, you will create a desktop access right, create a new role, assign the desktop right to the new role, and assign the role to Eval Group. At the end of this exercise, you will use the desktop right to modify a restricted folder. The steps in this exercise are similar to the steps that you performed in the preceding exercise to create an application right.

To create a role definition with a desktop access right:

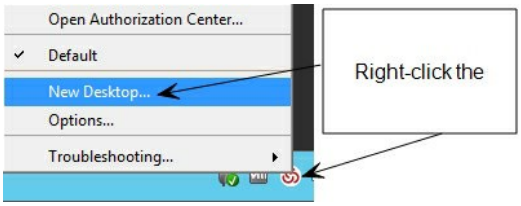
1. Log on with your administrator account and open Access Manager.
2. Create the new desktop right.
   - Select **Windows Right Definitions > Desktops**, right-click, then select **New Windows Desktop**.
   - Type DesktopRight as the name of the new desktop right on the General tab.
   - Click the **Run As** tab, then click **Add Built-in Groups**.
   - Select the **Administrators** group, then click **OK**.
   - Select **Re-authenticate current user** to require users to authenticate their identity when they use a role with this right, then click **OK**.
   - Select **Require multi-factor authentication** If you would like to enable multi-factor authentication for the right.

     Before you enable multi-factor authentication, you should be aware that multi-factor authentication for Delinea-managed Windows computers relies on the infrastructure provided by Privileged Access Service. For more information on preparing to use multi-factor authentication, see the *Multi-factor Authentication Quick Start Guide*.
3. Create a new role definition.
   - Select **Role Definitions**, right-click, then select **Add Role**.
   - Type DesktopAdmin as the name of the new role on the General tab.
   - Click **OK**.
4. Add the desktop right to the new role.
   - Select **Role Definitions**, right-click the DesktopAdmin role, and select **Add Right**.
   - Select DesktopRight and click **OK**.
5. Assign the role to a group.
   - Select **Role Assignments**, right-click, then select **Assign Role**.
   - Select DesktopAdmin from the list and click **OK** to display **Assign Role**.
   - Click **Add AD Account**.
   - Change the **Find** filter from User to Group.
   - Search for and select the group you created for the evaluation (for example, Eval Group), then click **OK**.
   - Verify that the account is included in the Accounts list in the Assign Roles dialog box, then click **OK**.
   - Open the **Privilege Elevation Service Settings** (from the Agent Configuration shortcut > Privilege Elevation Service > Settings), click the Troubleshooting tab, then click **Refresh** to get the latest authorization information.

To verify that the role with desktop rights grants elevated privileges:

1. Log off as the administrator and log on as amy.adams.
2. Open Windows Explorer and go to the C:\Windows folder.
3. Try to create a new folder in this location.

   From the default desktop for this account, the user does not have the necessary privileges to create a new folder. The only way she can create a new folder is by using administrator credentials.
4. Click the carat in the system tray notification area to display hidden icons, then click the Delinea icon to display the applet options.
5. Select **New Desktop**.



6. Select the DesktopAdmin role, then click **OK**.
7. Type the password for the logon account, then click **OK**.

   Notice that your new role is displayed when you left click on the Delinea icon in your task bar.
8. Try to create a new folder in the C:\Windows directory.

   Now you can create a new folder because the desktop that you are using has all of the rights associated with the Administrators group.

   Note: On Windows 10 and Windows Server 2016 systems, task bar menus are not available in an Elevated Desktop.

In this exercise, you created a role with the right to create a desktop with administrator privileges. You found that opening a new desktop with that role allowed you to perform administrative functions using your own credentials.

Switching Among Active Desktops

You can have multiple desktops available for you to use. For example, you might have separate desktop roles for managing Exchange and SQL Server that grant different rights. After you create a desktop for each role, you can switch between desktops by clicking the Delinea icon, then selecting the desktop you want to use. You can also set up hot keys to switch between desktops using a keystroke combination.

When you are finished working with a desktop, you can click the Delinea icon, then select **Close Desktop**.

Creating a Network Right

In the preceding section, you saw how you can provide a user with a desktop that has elevated privileges on a local computer (the Windows client computer in this case). However, using administrative privileges on your local Windows client computer does not give you privileges on a remote computer. In this section, you create a network access right that gives a user administrator privileges on a remote computer.

To illustrate network access rights using the local Windows client computer and a remote computer, you must install the Agent for Windows on the remote computer and join that remote computer to the zone you created in Creating the First Zone ("Creating the First Zone").

You can use the domain controller or another computer as the remote computer for this exercise. Install the Agent for Windows on the computer that you are using as the remote computer and join that computer to the Headquarters zone before proceeding.

If you are using only one Windows client computer for the evaluation and cannot install the agent on the domain controller or another remote computer, you should skip this exercise.

To prepare for the exercise that demonstrates this feature:

1. Install the Agent for Windows on the computer that you are using as the remote computer.

See Installing the Agent for Windows for more information.

2. Log on to the remote computer with your administrator account and create a folder on the C: drive named ShareFolder.
3. Select the folder, right-click, then select **Properties**.
4. Click the **Sharing** tab, then click **Share**.
5. Select Find people, type "back" to search for and select the built-in Backup Operators group, then click **OK**.
6. Right-click the Backup Operators group and set the Permission Level to Read/Write, then click **Share**.
7. Click **Done**, click **Close** to exit, then log off the remote computer as the administrator.
8. Log on to the local Windows client computer as amy.adams.
9. Try to open ShareFolder on the remote computer.
10. Verify that Windows tells you that you do not have sufficient permissions, then click **Cancel**.

To create a network access right and add it to the DesktopAdmin role:

1. Log on to the local Windows client computer with your administrator account and open Access Manager.
2. Create the new network access right.
   - Select *Windows Right Definitions > Network Access_*, right-click, then select **New Network Access**.
   - Type ShareAccess as the name of the new access right on the **General** tab.
   - Click the **Access** tab, select **Self with added group privileges**, then click **Add Built-in Groups**.
   - Select the **Backup Operators** group, then click **OK**.
   - Select **Re-authenticate current user** to require users to authenticate their identity when they use a role with this right, then click **OK**.
   - Select **Require multi-factor authentication** If you would like to enable multi-factor authentication for the right.

     Before you enable multi-factor authentication, you should be aware that multi-factor authentication for Delinea-managed Windows computers relies on the infrastructure provided by the Privileged Access Service. For more information on preparing to use multi-factor authentication, see the *Multi-factor Authentication Quick Start Guide*.
3. Add the new right to the existing DesktopAdmin role.
   - Under **Role Definitions**, select the DesktopAdmin role, right-click, then select **Add Right**.
   - Select the **ShareAccess** right in the list, then click **OK**.
4. Assign the role to a selected computer in the zone.
   - Expand the zone to **Computers > computer name> Role Assignments** node. If you are using a local and remote computer for this exercise, select the remote computer for making the role assignment.
   - Select **Role Assignments**, right-click, then select **Assign Role**.
   - Select DesktopAdmin in the list of roles, then click **OK**.
5. Assign the role to an Active Directory group.
   - Click **Add AD Account**.
   - Change the **Find** filter from User to Group to search for and select the group you created for the evaluation (for example, Eval Group), then click **OK**.
   - Verify that the account is included in the Accounts list in the Assign Roles dialog box, then click **OK**.
   - Open the **Privilege Elevation Service Settings** (from the Agent Configuration shortcut > Privilege Elevation Service > Settings), click the Troubleshooting tab, then click **Refresh** to get the latest authorization information.

To verify the role with network access rights grants elevated privileges:

1. Log on to the local Windows client computer as amy.adams.

   If you try to open ShareFolder in the default desktop, Windows denies access.

2. Open the applet, select **New Desktop**, and select the DesktopAdmin role.

   This role has the network access right that gives you remote access to the computer running as the account with Read/Write permission.

3. Open ShareFolder and verify that Windows gives you access.

In this exercise, you added a remote access right to a role that already had a desktop right and saw how changing desktops changes the user's rights.

Reviewing Rights and Roles in the Authorization Center

The Authorization Center is an option available from the applet menu. You can use the Authorization Center to display detailed information about your currently available rights and role assignments.

To view the Authorization Center:

1. Click the Delinea icon in the notification area.
2. Select **Open Authorization Center**.
3. Click through the tabs to view detailed information about your rights, roles, role assignments, and auditing status.

If you have completed all the steps in the preceding chapters, the audit and monitoring service has been auditing your sessions as an administrator and the user account you created. This chapter describes how to view audited sessions, update the review status, and use queries to find the sessions in which you're interested.

Using Audit Analyzer to Replay a Session

If you selected both Privilege Elevation Service and Auditing and Monitoring Service features, the Agent for Windows has been capturing your activity as you logged on and off and switched between roles. You can replay those recorded sessions to see detailed information about what you did during the evaluation. Before you can replay the sessions captured, however, you use Audit Analyzer to locate the sessions you are interested in using a set of predefined queries. For example, there are predefined queries for sessions that started Today and This Month and sessions where the Windows Command Prompt or Windows MMC tools were used.

To select and replay a session:

1. Open Audit Analyzer to view captured sessions grouped by predefined queries.
2. Select a predefined query, such as **Today** or **Active Sessions**, in the left pane to display a list of sessions in the right pane.

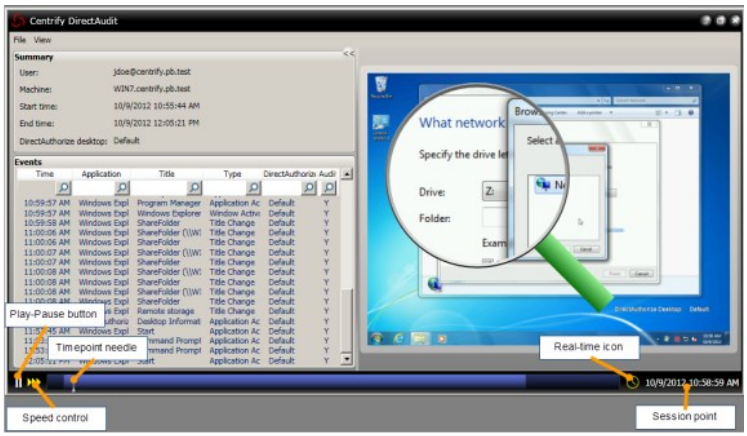
   Note that the date queries show sessions that started during the specified time interval. If a session started three days ago and is still active, it is listed under This Week and Active Sessions, but not under Today or Yesterday.
3. Double-click a session to retrieve the session from the database and open the session replay window.

   The session replay window displays information similar to the following:

   The replay progress is shown in the play bar along the bottom of the window. If you double-click an event, you can watch the recording of just that event.

Magnifying the Recorded Session

You can click the magnifier in the replay window to enlarge a portion of the recorded screen. The magnifier appears as a magnifying-glass pointer in the replay pane. Click to toggle the magnifier on or off.



Controlling Playback Speed or Session Location

For normal playback operation, you can click Play or Pause to start or pause a session. You can also fast forward by clicking the Speed control. The Timepoint needle shows you the current location in the session. You can drag the needle to any point in the session. The Real-time icon to the right of the time bar indicates that the session plays in a smooth time sequence. If you want to play back the session moving swiftly from one user action to the next, click the icon to gray it out. The Session point indicates the date and time of the Timepoint needle.

Marking Sessions for Review or Action

You can use Audit Analyzer to manage the status of sessions that are pending review or action. For example, you can update the status of individual sessions using the following states:

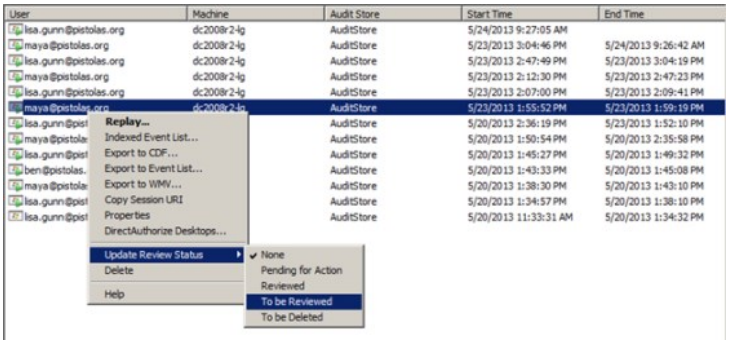
- To be Reviewed
- Reviewed
- Pending for Action
- To be Deleted

After you have marked sessions to be reviewed or pending action, you can use the predefined queries **Sessions to be Reviewed** and **Sessions Pending for Action** to see only the sessions in those states.

To update the review status for a session:

1. Select a query then select an individual session.
2. Right-click and select **Update Review Status**, then select a review state.

   For example, if the session is new and has not been reviewed, select **To be reviewed**.



3. Type a comment at the prompt, then click **OK**.
4. Click **Sessions to be Reviewed** in the left pane to see the session displayed.

   You can also view the review status and comments for a session by right-clicking a session, then select Properties.
5. Select one or more sessions and update the review status to Reviewed.

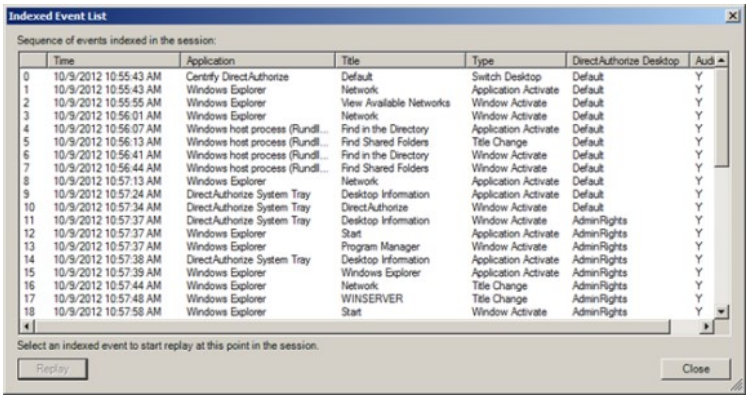
   Again, you will be prompted to provide a comment for the change in status. Type a new comment and click OK.

Using the Indexed Event List

If you don't want to replay an entire session, you can use the indexed event list to view a summary of events recorded in a session, then selective start the replay at a specific event of interest.

To use the indexed event list:

1. Select a query then select an individual session.
2. Right-click and select **Indexed Event List**.



3. Select a session event in the lists to start the replay at that event.

Creating Custom Queries

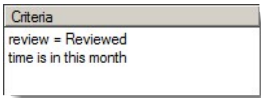
Predefined queries searches the audit store database for sessions that meet the specific criteria. To see the search criteria, right-click a query, select **Properties**, then click the Definition tab.

You can write your own queries to search for sessions that meet specific criteria of your choosing. The following example illustrates how to build a query that finds all of the sessions that have been reviewed.

To create a custom query for sessions that have been reviewed:

1. Open Audit Analyzer.
2. Select **Audit Sessions**, right-click, then select **New Shared Query**.
3. Type Reviewed Sessions for the name of the query and enter a description for the query. For example, type Sessions that have been reviewed by department auditors.
4. Deselect UNIX session as the type of session to include.
5. Click **Add** to add criteria.

   Notice that review = Reviewed appears in the Criteria field of the New Query dialog box.
6. Select **Review Status** from the Attribute list, select **Reviewed**, then click **OK**.
7. Click **Add** again.
8. Select **Session Time**, select the bottom radio button and **Is in**, then select **this month** and click **OK**.
9. Verify the Criteria displays both rules, then click **OK** to complete the query.

   After you click OK, the query is listed under **Shared Queries**.
10. Click the custom query to get the results.

Creating a Quick Query

You can also perform quick text string searches in Audit Analyzer.

To create a quick text string query for sessions:

1. Open Audit Analyzer.

2. Select **Audit Sessions**, right-click, then select **New Quick Query**.
3. Type a search string into the dialog box.

   As you type, the Quick Query displays a list of possible matches that start with the text you are typing. If an item in the list is what you are looking for, select it, then click **Find** to display all matching sessions in the right pane.

Auditing Only Specific Events

The integration of access management and auditing makes it possible for you to audit only when a user switches to a specific desktop or role. Although you can use database queries in Audit Analyzer to find recorded events of a particular type, you can save space in your database by recording only those events in which you're most interested.

Specifying which Roles or Desktops to Audit

To limit auditing to specific roles or desktops, you turn off more generalized auditing and enable auditing for just the roles you care about. The following example illustrates how to audit only when the user switches to a privileged desktop.

To audit only when the user switches to a privileged desktop:

1. Log in to the computer as the Administrator and open Access Manager.
2. Expand the console tree to the Authorization node for your evaluation zone.
3. Expand Role Definitions, select the DesktopAdmin role, right-click, then select **Properties**.
4. Click the Audit tab, select **Audit if possible** or **Audit required**.If auditing is required, users are prevented from using the role if auditing is not available or the agent is not running.
5. Log off and then log in as amy.adams.
6. Verify that you do not have elevated privileges by trying to change firewall settings in Control Panel.
7. Open a new desktop and select the DesktopAdmin role.
8. Perform operations, such as running the Firewall Control Panel and accessing the remote share on the Windows server, for which you need elevated privileges.
9. Switch back to your default desktop.
10. Open Audit Analyzer, select the Active Sessions node, and refresh the display.
11. Open the currently active session for the Windows client computer.

    You should find that only the portion of the session when you were using the DesktopAdmin desktop was recorded.

Audit Trail of Privileged Events

Even when the auditing and monitoring service is not recording a session, it keeps a record of every event in which the user selected a role that provides elevated privileges.

To view audit trail events for elevated privileges:

1. Log in using your administrator account and open Access Manager.
2. Expand the console tree to the Authorization node for your evaluation zone.
3. Select the **ControlPanelAdmin** role, right-click, then select **Properties**.
4. Click the **Audit** tab and select **Audit not requested/required**.
5. Log off and then log in as amy.adams.
6. Verify that you do not have elevated privileges by trying to change firewall settings in Control Panel.

7. Right-click your Control Panel shortcut, select the **ControlPanelAdmin** role, and verify that you now have the rights to change firewall settings.
8. Close Control Panel and perform several more operations.
9. On the Windows client computer, open **Audit Analyzer**, select **Active Sessions**, and refresh the display.
10. Open the currently active session for your Windows client computer. You should find that none of your recent operations were recorded.
11. Right-click the Audit Events node, then select **Query Audit Events**.
12. In the dialog box, enter your search criteria, such as a role name, event time, or the type of event you are interested in locating, then click **OK**. All of the events that match the criteria you specify are listed. If the event involved an audited role and you are capturing video records of audited activity, you can right-click an event to **Replay** the activity recorded.

    All of the events that match the criteria you specify are listed. If the event involved an audited role and you are capturing video records of audited activity, you can right-click an event to **Replay** the activity recorded.

Additional Auditing Tools

Because the evaluation computer has the complete auditing infrastructure, you have several additional tools available for managing different components of that infrastructure. For example, computers that have the Agent for Windows installed also have the following Auditing and Monitoring Service Settings. You also have access to the Audit Collector Control Panel, Audit Management Control Panel, and Audit Manager console. All of these programs are available from the Windows Start menu.

You use the control panels to configure and troubleshoot the component operations. Audit Manager provides a overview of all audit-related components. From Audit Manager, you can view the status of components, modify component properties and relationships, and manage audit store databases. You can also use Audit Manager to create audit roles, assign users to the audit roles, and manage permissions.

Audit Manager includes one Master Auditor role with full control over the installation. As the Master Auditor, you can manage and control all permissions for the installation.

The *Server Suite Evaluation Guide for Linux and UNIX* describes how to install and configure the Server Suite software on a Windows computer joined to an Active Directory domain controller and on the Linux and UNIX computers you want to manage. After you install the software, you can follow the steps in this guide to create Active Directory users and groups and set up a test environment with Server Suite zones, roles, privileges, and group policies. Through this test environment, you can see how Server Suite enables you to control users access, manage privileges, and monitor activity on UNIX and Linux computers in your organization.

Intended Audience

This guide is for system and network administrators who want to evaluate Server Suite software. The guide assumes you have a working knowledge of Windows Server and Active Directory and are familiar with Active Directory features, functionality, and terminology. This guide also assumes you are familiar with the Linux or UNIX-based computers you plan to manage and how to perform common administrative tasks.

Using this Guide

Server Suite provides an integrated set of software components that centrally control, secure, and audit user access to servers, workstations, mobile devices, and applications through Microsoft Active Directory. The purpose of this guide is to give you hands-on experience using Server Suite software to manage identities, access privileges, and administrative tasks on UNIX and Linux computers.

The guide is divided into the following chapters:

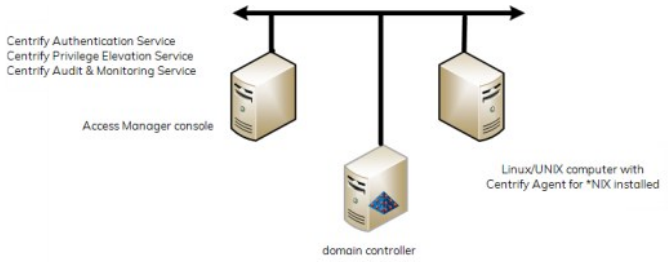
- Preparing Hardware and Software for an Evaluation describes what you will need and how to prepare for the evaluation.
- Configuring the Basic Evaluation Environment provides step-by-step instructions for setting up the evaluation environment.
- Exploring Additional Management Tools describes the features of Server Suite software that reduce complexity and ease the workload in large organizations.
- Auditing Sessions describes how you can audit user activity and search and replay user sessions.
- Frequently Asked Questions provides answers to the most common questions about Server Suite products and features.
- Removing Software after an Evaluation describes how to optionally uninstall Server Suite software.

Preparing Hardware and Software for an Evaluation

This chapter describes the hardware and software you need to prepare for the evaluation of Server Suite software. It includes instructions for downloading SErver Suite software from the Delinea website if you do not have the CD and the permissions required to install and configure the evaluation environment.

What You Need for the Evaluation

To follow the instructions in this guide, you need a simple configuration of networked Windows domain computer, Windows Server domain controller, and a Linux, UNIX, or Mac OS X computer to manage as illustrated in the following example.



To complete this evaluation, you install Server Suite software on two physical or virtual computers:

- **Authentication & Privilege** and **Audit** components on a Windows computer joined to an Active Directory domain.
- **Server Suite Agent for *NIX** on a supported Linux-based or UNIX-based platform that you want to manage.

In most organizations, Server Suite software is not installed on the domain controller. However, you must be able to connect to a domain controller from the other two computers to complete the evaluation.

- Windows Computer Requirements
- Linux and UNIX Computer Requirements
- Domain Controller Requirements

Windows Computer Requirements

You use the Windows computer where Access Manager is installed to perform most of the procedures described in this guide.

Before installing on Windows, check that you have a supported version of one of the Windows operating system product families.

For details about supported platforms, please consult the release notes.

You should also verify that you have the .NET Framework, version 4.5 or later, installed. If the .NET Framework is not installed, the setup program can install it for you. Alternatively, you can download the .NET Framework from the Microsoft Download Center, if needed.

The Windows computer should have the following minimum hardware configuration:

CPU speed	550 MHZ
RAM	256 MB
Disk space	1.5 GB

You should also verify that the Windows computer you plan to use for the evaluation is joined to the Active Directory domain.

> **Note**: If you are installing the software on virtual computers, see Using a Virtual Environment for additional guidelines.

Linux and UNIX Computer Requirements

A platform-specific Server Suite Agent for *NIX must be installed on each computer you want to manage through Active Directory. Delinea supports several hundred distributions of popular operating systems, including AIX, HP-UX, and Solaris versions of the UNIX operating environment and both commercial and open source versions of the Linux operating system. For the most complete and most up-to-date list of supported operating systems and vendors, see the supported platforms listed in the release notes.

You can download platform-specific agent packages from the Delinea **Customer Download Center** if you register for a free account. You can also download agents for free from the Centrify Express website.

The UNIX or Linux computer must be connected to the same network as the domain controller.

Domain Controller Requirements

For the Active Directory domain controller, you should verify that you have access to a computer with a supported version of the Windows Server product family and is configured with the domain controller and DNS server roles. For details about supported platforms, please consult the release notes.

In addition, you should verify that the domain functional level is at least Windows Server 2008 R2.

To determine the domain functional level

1. Open **Active Directory Users and Computers** (dsa.msc).
2. Select the domain.
3. Select **Action**, then click **Raise domain functional level**.

   If the current domain functional level is not at least Window Server 2008 R2, use the drop down list to raise the level.

Verifying Administrative Access for the Evaluation

To prepare for the evaluation, you should confirm that you have the local Administrator account and password for the root domain of the Active Directory forest. The forest root Administrator account is the account created when you install the first Windows Server in a new Active Directory site.

If you set up a separate Active Directory domain for testing purposes, you should have this account information. If you are using an existing Active Directory forest that was not expressly created for the evaluation, you should identify the forest root domain and confirm that you have an account that is a member of the Domain Admins group on the Windows computer you use for the Access Manager console. This ensures that you have all the permissions you need to perform the procedures in this evaluation.

If you are not a member of the Domain Admins group on the Windows computer you use for the Access Manager console, have the Active Directory administrator create a separate organizational unit for Delinea objects and delegate control of that organizational unit to the user account you are using for evaluation. For more information about delegating control, see Delegating control for the Delinea organizational unit.

You should also verify that Administrative Tools are visible in the Start menu on the Windows computer you are using for the evaluation. If the Administrative Tools option is not displayed, download and install the Microsoft Remote Server Administrator Tools from the Microsoft website. For download and installation instructions, see http://www.microsoft.com/en-us/download/details.aspx?id=7887.

Checking the DNS Environment

The Server Suite Agent is designed to perform the same set of DNS lookups that a typical Windows computer performs in order to find the nearest domain controller for the local site. For example, the Server Suite Agent for *NIX looks for service locator (SRV) records in the DNS server to find the appropriate controller for the domain it has joined.

In most cases, when you configure the DNS Server role on a Windows computer, you configure it to allow dynamic updates for Active Directory services. This ensures that the SRV records published when a domain controller comes online are available in DNS. If your DNS Server is configured to prevent dynamic updates, however, or if you are not using the Window computer as the DNS server, the Server Suite Agent for *NIX might not be able to locate the domain controller.

Do the following to ensure the UNIX computer can look up the SRV records in the DNS server for the evaluation environment:

- Configure the DNS Server role on the Windows computer to **Allow secure dynamic updates**.
- Make sure that each UNIX or Linux computer you are using includes the Windows DNS server as a nameserver in the /etc/resolv.conf file.

When you configure the DNS Server, you should configure it to perform both forward and reverse lookups and to allow secure dynamic updates.

Using a Virtual Environment

To simplify the hardware requirements, you might find it useful to set up your evaluation environment using either Microsoft Virtual PC or VMware Workstation. To set up a virtual environment, you need a computer with enough CPU, RAM, and available disk space to run three virtual machines simultaneously. Delinea recommends the following minimum configuration:

- CPU: at least 1.70 GHz
- RAM: at least 8 GB
- Available disk space: 15 GB

The virtual environment should also be configured to run as an isolated evaluation environment using **Local/Host-only** or **Shared/NAT** networking.

In addition, because the virtual environment runs as an isolated network, each virtual machine should be manually assigned its own static TCP/IP address and host name.

Downloading Server Suite Software for UNIX Evaluations

You can go to the Centrify website to sign up for a free trial. Once you're signed up with an account, you can download the software.

To register for a Delinea free trial

1. Navigate to https://www.centrify.com/free-trial/.
2. Enter your contact and company information, click the checkbox to indicate that you agree to the terms of use and privacy policy, and then click **Start My Trial**.

   Note: You will receive an email with the next steps in downloading your free trial.

Downloading Server Suite Windows Software

You can download all of the components for Server Suite from the Delinea website to your Windows computer. Before you begin, be sure you have the email address and password you used to register for your trial.

To download the Windows software for Server Suite

1. Open a browser on the Windows computer you plan to use for the evaluation and go to www.centrify.com.
2. In the upper area of the web page, click **Login**.
3. Enter your email address and your account password, then click **Login**.
4. Go to **Support** › **Downloads**.
5. Select **Zero Trust Privileges - Enterprise** to locate the latest software bundles.
6. Next to the latest version for 64-bit Windows systems, click either the **ISO** or **ZIP** button to download the software in that format.

   The latest version of the Windows software bundle is called Server Suite.
7. Close the window when the download is complete.

Downloading the Linux and UNIX Agents

You can download individual platform-specific packages directly from the Delinea website to a local Linux or UNIX computer.

To download UNIX and Linux agent packages

1. Open a browser on the Linux or UNIX computer you plan to use for the evaluation and go to www.centrify.com.
2. In the upper area of the web page, click **Login**.
3. Enter your email address and your account password, then click **Login**.
4. Go to **Support** › **Downloads**.
5. If you want the bundle that has all of the UNIX/Linux agents:
   1. Select **Zero Trust Privileges - Enterprise** to locate the latest software bundles.
   2. Next to the Agents for UNIX/Linux - All-in-One disk, click either the **ISO** or **ZIP** button to download the software in that format.
6. If you want the agent package just for your specific UNIX/Linux system:
   1. Select **Authentication Service** to locate the latest software bundles for the various *NIX systems.
   2. Next to the Agent for your preferred operating system, click the **TGZ** button to download the software in that format.

Verifying that You Have Active Directory Permissions

Many of the procedures in this guide add or modify Active Directory user, group, and computer accounts. You should verify you have the appropriate Active Directory permissions to make these kinds of changes in the evaluation environment. If you are not an Active Directory administrator or a domain administrator, you might not have access to the domain controller or sufficient permission to modify Active Directory objects and attributes.

To conduct the evaluation, have an Active Directory administrator create an organizational unit for you to use and delegate full control of the organizational unit to you. For more information about creating an organizational unit and delegating control, see the following topics:

- Creating an organizational unit for Delinea
- Delegating control for the Delinea organizational unit

In addition to the organizational unit for Delinea objects, you need to have **Log on as a service** user access rights to start the Zone Provisioning Agent included in the package.

To confirm that your account has "Log on as a service" access rights

1. Open the **Windows Administrative Tools Local Security Policy**.
2. Expand the **Local Policies** node and select **User Rights Assignments**.
3. Scroll down to **Log on as a service** and double-click to display properties for this right.
4. Click **Add User or Group**.
5. Type the user or group name or click **Browse** to search for and select your account, then click **OK** to add this right to your account in the Local Security Setting.

Next Steps

This concludes the site preparation, Server Suite software download, and permissions assessment. You are now ready to install the software and create the fundamental elements of the evaluation environment.

Configuring the Basic Evaluation Environment

In this chapter, you install Server Suite software on your evaluation computers and configure users, groups, roles, and group policies to integrate the UNIX environment into Active Directory. After you complete these steps, your UNIX or Linux computer will be a Server Suite-managed computer that is joined to the Active Directory domain, allowing UNIX users to log in using their Active Directory credentials.

Complete the tasks in order, as described in the following sections.

Creating an Organizational Unit

To isolate the evaluation environment from other objects in Active Directory, you can create a separate organizational unit for all of the Delinea-specific objects that are created and managed throughout the evaluation. You must be the Active Directory administrator or have Domain Admins privileges to perform this task.

To create an organizational unit for Delinea

1. Open **Active Directory Users and Computers** and select the domain.
2. Right-click and select **New > Organizational Unit.**
3. Deselect **Protect container from accidental deletion.**
4. Type the name for the organizational unit, for example, Delinea, then click **OK**.

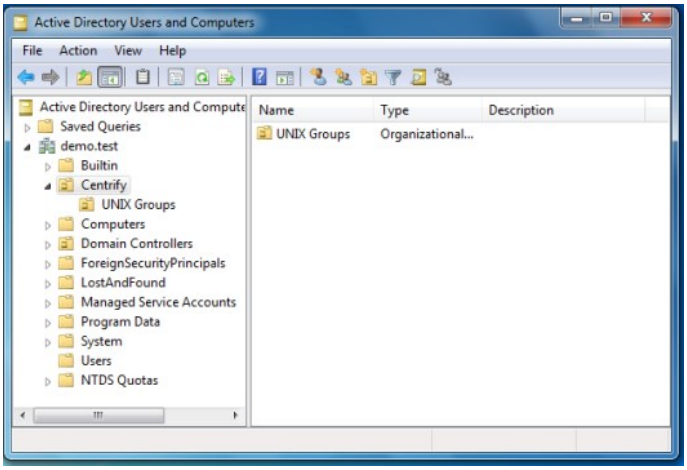
Create Additional Organizational Units

Additional organizational units are not required for an evaluation. In a production environment, however, you might create several additional containers to control ownership and permissions for specific types of Delinea objects. For example, you might create separate organizational units for UNIX Computers and UNIX Groups.

To illustrate the procedure, the following steps create an organizational unit for the Active Directory groups that will be used in the evaluation to assign user access rights to the Delinea-managed computers within the top-level organizational unit for Delinea-specific objects.

To create an organizational unit for evaluation groups

1. In **Active Directory Users and Computers**, select the top-level organizational unit you created in Creating an organizational unit for Delinea.
2. Right-click and select **New > Organizational Unit.**
3. Deselect **Protect container from accidental deletion.**
4. Type the name for the organizational unit, for example, UNIX Groups, then click **OK**.



In later exercises, you will use this organizational unit and add other containers to manage additional types of information.

Delegating Control for the Organizational Unit

To allow another person who is not an Active Directory administrator to perform all of tasks in the evaluation, you can delegate control of the Delinea organizational unit to that person. If you are an Active Directory administrator or a member of the Domain Admins group in the evaluation domain, you can skip this step.

To delegate control of the organizational unit for Delinea

1. Open **Active Directory Users and Computers** and select the domain.
2. Select the top-level organizational unit for Delinea objects, Delinea.
3. Right-click, then select **Delegate Control**.
4. In the Delegation of Control wizard, click **Next**.
5. Click **Add**.
6. Search for and select the user or group for delegation, then click **Next**.
7. Select the tasks to delegate, then click **Next**.

   At a minimum, select the following common tasks:

   - Create, delete, and manage user accounts
   - Reset user passwords and force password change at next logon
   - Read all user information
   - Create, delete, and manage groups
   - Modify the membership of a group
8. If you are delegating the task of joining computers to a zone, you can specify the scope of computers you can join to the zone; you pick a container in Active Directory to grant access to.

   If you leave the scope blank, the scope is the domain root. Be aware that the postalAddress field is used for information about joining computers to a zone; if you lookup the permissions for people you've delegated the task of joining computers to a zone, they'll have permissions to the postalAddress field for the affected computers.
9. Click **Finish.**

Installing and Configuring Access Manager

You are now ready to install Access Manager and other components on the Windows computer you are using for the evaluation.

To install components on the Windows computer

1. On the physical or virtual computer where you downloaded Server Suite software, double-click **autorun.**
2. On the **Getting Started** page, click **Authentication & Privilege.**
3. On the **Welcome** page click **Next**.
4. Review the terms of the license agreement, click **I agree to these terms**, then click **Next**.
5. Type your name and organization, then click **Next**.
6. Select the components to install, then click **Next**.
7. Accept the default **C:\Program Files\Centrify** location for installing components, or click **Browse** to select a different location, then click **Next**.
8. Click **Next** to disable publisher verification.
9. Review the components you have selected, then click **Next** to begin installing components.
10. Deselect the Configure and start Zone Provisioning Agent option, then click **Finish**.

    Because you are going to configure the service account for the Zone Provisioning Agent in a later exercise, click **Yes** to dismiss the warning about the Zone Provisioning Agent running as the local system account.
11. Click **Exit** to close the Getting Started page.

Starting Access Manager for the First Time

After installing Access Manager and other components, you should have the new Access Manager icons on your desktop.

You are now ready to start using Access Manager. The first time you open Access Manager it creates Active Directory containers to store Delinea licenses and zone information.

To start Access Manager for the first time

1. Open **Access Manager** by double-clicking the icon on the desktop.
2. Verify the name of the domain controller, then click **OK**.

    The default is the domain controller to which the Windows computer is joined. If you want to connect to a different forest, type the name of a domain controller in that forest. If you want to connect to the forest with different credentials, select **Connect as another user**, then type a user name and password to connect as.
3. In the Setup Wizard Welcome page, click **Next**.
4. Verify that **Use currently connected user credentials** is selected to use your current logon account, then click **Next**.

    You must be logged on with an account that has Active Directory administrator rights in the target organizational unit.

    If your logon account does not have those rights, select **Specify alternate user credentials** and enter a different user name and password.
5. Select **Generate Delinea recommended deployment structure** and **Generate default deployment structure**, then click **Next**.
6. Select a location for installing license keys in Active Directory, then click **Next**.

    The Setup Wizard displays information about the Read permissions that must be granted on the container. Click **Yes** to continue.
7. Type or copy and paste the license key you received, click **Add**, then click **Next**.

    If you received the license key in a text file, you can click **Import** to import the key directly from the file, then click **Next**.

8. Click **Next** to use the default container for the Delinea zones.
9. Accept the default permission delegation and click **Next**.
10. Review the summary of your selections, then click **Next**.
11. Click **Finish**.

    After you click Finish, Access Manager displays.

Creating the First Zone

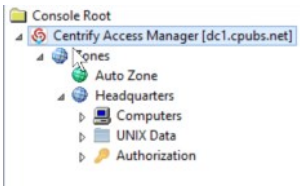
The next step in configuring your evaluation for access control and privilege management is to create a Delinea zone. Zones enable you to define and control access privileges for users and groups in your organization. By using zones, you can limit who has access to different computers and where users have permission to exercise elevated privileges.

To create a parent zone

1. Open **Access Manager**.
2. Click **Create Zone**.

3. Type a name and description for the zone, for example Headquarters, then click **Next**.
4. Leave **Use default zone type** selected, and click **Next**.
5. Verify information about the zone you are creating, then click **Finish**.

You now have one parent zone. You can have multiple parent zones or a single parent zone, depending on your needs. If you expand the **Zones** node, the left pane displays your new zone.



Access Manager automatically creates the Computers, UNIX Data and Authorization nodes for each zone you create. These nodes enable you specify precise access privileges for computer and application administrators in each zone.

A parent zone can have one or more child zones. Child zones inherit information from the parent zone. For example, you can define access rights, roles, and role assignments in a parent zone and use them or change them in a child zone. You will work with child zones in a later exercise.

Now that you have Access Manager installed and have configured your first zone, you are ready to install the Server Suite Agent on a UNIX or Linux computer.

Installing the Server Suite Agent for

The Centrify Agent must be installed on each UNIX or Linux computer you want to manage. After you have downloaded platform-specific agents for the operating systems you want to evaluate, you should make sure the software is on the physical or virtual UNIX or Linux computer you are using for the evaluation.

To install the agent package

1. Log on to the UNIX or Linux computer with root privileges.
2. Copy the Centrify Agent for *NIX package for the local operating system to the computer and change to that directory.
3. Extract the contents of the package.

   For example, if you have a Red Hat Enterprise Linux based computer, you might enter the following:

   gunzip centrify-server-suite-‹*release*›-rhel5-x86_64.tgz
4. Expand the archive file.

   For example, if you have a Red Hat Enterprise Linux based computer, you might enter the following:

   tar -xvf centrify-server-suites-‹*release*›-rhel5-x86_64.tar
5. Run the install.sh script.

   For example, if you are running Red Hat Enterprise Linux you would enter the following:

   /bin/sh install.sh
6. Follow the prompts displayed to check whether the local computer is ready for the installation.

   If there are errors, you must fix them before installing the software. Warning messages are informational, but do not prevent you from installing the software.
7. Follow the prompts displayed using the following instructions:

Do you want to run adcheck to verify your AD environment?	Enter N to skip post-installation checks.
Join an Active Directory Domain?	Enter N to join later.
Enable auditing on this computer (audit and monitoring service NSS mode)?	Enter Y to enable auditing.
Do you want to continue (Y) or re-enter information?	Enter Y to install the default packages.
Enable Linux Desktop auditing on this computer?	Enter Y to enable Linux desktop auditing.

If you have more than one Linux or UNIX computers included in the evaluation, repeat Step 1 through Step 7 on each computer.

8. Verify the installation by running the adinfo command at the UNIX command prompt.

   adinfo

   This command-line program displays information about the Linux or UNIX computer's status in Active Directory. At this point, the output should show you that you are not joined, but Licensed Features are enabled.

Joining the Domain

You are now ready to use the adjoin command-line program to join the Linux or UNIX computer to the Active Directory domain you are using for evaluation.

The most basic syntax for the adjoin command is:

```
adjoin domain -z zone -u username
```

For more information about adjoin syntax and options, see the man page for the adjoin command.

To join an Active Directory domain from a Linux or UNIX computer

1. Log on to the UNIX or Linux computer with root privileges.
2. Run the adjoin command, specifying the domain, zone, and the account name for an Active Directory administrator with permission to join the domain.
3. Enter the password for the Active Directory account used to join the domain.
4. Verify the UNIX or Linux computer is joined to Active Directory by running the adinfo command.

   adinfo

   The output should look similar to the following:

   ```
   Local host name: my-eval
   Joined to domain: test.acme.com
   Joined as: my-eval.test.acme.com
   Pre-win2K name: my-eval
   Current DC: dc-mine.test.acme.com
   Preferred site: CA
   Zone: test.acme.com/acme/zones/HQ
   Last password set: 2020-08-14 11:24:32 PDT
   CentrifyDC mode: connected
   Licensed Features: Enabled
   ```

5. Restart the Linux or UNIX computer.

   Restarting the computer is not required, but is recommended to ensure that all services are restarted.

Verifying your Progress in Access Manager

You now have a Server Suite-managed computer. To see the computer in Access Manager, expand **Zones > Headquarters > Computers**. The Linux or UNIX computer is listed under the Computers node. The computer has successfully joined an Active Directory domain and is prepared for access control and privilege management. However, no Active Directory users can log on to the computer yet.

Adding and Provisioning an Evaluation User and Group

Before any Active Directory users can log on to the Server Suite-managed computer, you must provision an Active Directory account with UNIX profile attributes and assign the user a role that has login privileges. To demonstrate the process in the evaluation, you will create a new Active Directory user, provision the user with a UNIX profile, and assign the user basic access privileges.

To create a new Active Directory user with access to the Server Suite-managed computer

1. Open **Active Directory Users and Computers** and create a new **User** object.
    1. Fill in the **First**, **Last**, and the **User logon** name fields.
    2. Type and confirm a password and select the **Password never expires** option.
    3. Acknowledge the warning, click **Next**, then click **Finish**.
2. Create a new Active Directory group in the UNIX Groups organizational unit you created under the Delinea organizational unit.
    1. For the **Group name** enter Login Users.
    2. Select **Global** as the scope for the group and **Security** for the type of group, then click **OK**.
3. Add the evaluation user to the Login Users group.
    1. Select the user you created in Step 1, right-click and select **Add to a group**.
    2. Select the **Login Users** group, then click **OK**.
4. Provision a UNIX profile for the new user using Access Manager.
    1. Expand the Zones node and select the Headquarters, right-click, then select **Add User.**
    2. Select the user you created for the evaluation.
    3. Select **Define user UNIX profile only** and deselect **Assign roles**.
    4. Accept the default values for all profile properties.
    5. Review your selections, click **Next**, then click **Finish**.
5. Assign the default UNIX Login role to the Login Users group using Access Manager.
    1. Expand the **Authorization** node under the Headquarters zone.
    2. Select **Role Assignments**, right-click, then select **Assign Role**.
    3. Select the **UNIX Login role** and click **OK**.
    4. Click **Add AD account**.
    5. Change the object to **Find from User to Group**, then search for and select the Login Users group, then click **OK**.
    6. Click **OK** to complete the role assignment.

Verify Access by Logging On

The Active Directory user can now log on to the UNIX or Linux computers that has joined the domain and the parent zone.

To verify the user can log on using Active Directory credentials

1. Open a terminal on your joined Linux or UNIX computer and switch to the root account.
2. Run adflush to clear the Server Suite Agent for *NIX's cache.

    This step simply ensures that the agent will make a new connection to Active Directory to get the latest user and group information.

3. Log off as root.
4. Log in using the Active Directory credentials for the evaluation user you created and added to the Login User group.

Creating a UNIX Administrator Role

Now that you have verified an Active Directory user can access the Linux or UNIX computer you are using for the evaluation, you will see to how to create users that have elevated privileges and how you can limit the use of those privileges to specific computers.

To illustrate this scenario, you will create a UNIX administrator role that grants root privileges for the computers in a zone without requiring users to know the root password. Instead, users who are assigned the UNIX administrator role use their Active Directory credentials.

You can use the same steps to define roles with different and more granular rights. For example, you will follow similar steps to create an Apache administrator role that can only perform a limited set of tasks on computers in a child zone.

At the end of this section, you will have two accounts with UNIX Login privileges: one of which has only standard user privileges, the other account has full administrative privileges.

To create a new Active Directory user and group with administrative access

1. Open Active Directory Users and Computers and create a new **User** object.
    1. Fill in the First, Last, and the User logon name fields.
    2. Type and confirm a password and select the Password never expires option.
    3. Acknowledge the warning, click Next, then click Finish.
2. Open Active Directory Users and Computers and create a new **Group** object in the UNIX Groups organizational unit.
    1. For the Group name, enter EnterpriseUnixAdmins.
    2. Select Global as the scope for the group and Security for the type of group, then click OK.
3. Add the administrative user to the EnterpriseUnixAdmins group.
    1. Select the user you created in Step 1, right-click and select Add to a group.
    2. Select the EnterpriseUnixAdmins group, then click OK.
4. Provision a UNIX profile for the new user using Access Manager.
    1. Expand the Zones node and select the Headquarters, right-click, then select **Add User.**
    2. Select the user you created for UNIX administration.
    3. Select Define user UNIX profile only and deselect Assign roles.
    4. Accept the default values for all profile properties.
    5. Review your selections, click Next, then click Finish.

Defining a Command Right and a New Role

You are now ready to define a new privileged command right that uses the asterisk (*) wild card to give the user the equivalent of all commands, all paths, and all hosts in the sudoers file. In a production deployment, you would define more specific sets of privileged commands and run them using accounts with no restricted access than the root user.

To create new UNIX right definition for the administrative role

1. Create a new privileged command using Access Manager.
    1. Expand the Authorization node under the Headquarters zone, then expand UNIX Right Definitions and select Commands.
    2. Right-click then select New Command. For this example, you will only set information on the General tab.
    3. Type a command name and description, for example root_any_command and All commands, all paths, all hosts.
    4. Type an asterisk (*) in the Command field to match all commands.

5. Leave the default setting for Glob expressions.
6. Select the Specific path options and type an asterisk (*) to match all command paths, then click OK.

You now have a root_any_command that grants privileges to run any command in your role definitions. In the next steps, you create a role that will give members of the EnterpriseUnixAdmins group the root_any_command privileges.

To create and assign the UNIX administrators role

1. Create a new role definition using Access Manager.
    1. Expand the Authorization node under the Headquarters zone, select Role Definitions, right-click, then select **Add Role**.
    2. Type a role name (UnixAdminRights) and a description (Set of rights for UNIX administrators) for the new role.
    3. Click the **System Rights tab** and select all of the UNIX rights and the Rescue right.
    4. Click the Audit tab and select Audit if possible, then click **OK**.
2. Add the root_any_command and several default rights to the new role.
    1. Select the UnixAdminRights role, right-click, then select **Add Right**.
    2. Use CTRL-click to select rights, including login-all, secure shell (ssh, sshd, and dzsshall) rights, and the root_any_command right you just created, then click **OK**.
3. Assign the UnixAdminRights role to the enterprise UNIX administrators group using Access Manager.
    1. Expand the Authorization node under the Headquarters zone, select Role Assignments, right-click, then select Assign Role.
    2. Select the UnixAdminRights role and click **OK**.
    3. Click **Add AD Account**.
    4. Change the object to Find from User to Group, then search for and select the **EnterpriseUnixAdmins group**, then click **OK**.
    5. Click **OK** to complete the role assignment.

Verifying Administrative Privileges

You now have two role assignments—Login Users and EnterpriseUnixAdmins—in the zone. Any Active Directory user you add to the Login Users group and provision a UNIX profile for will have access rights but no administrative privileges on the computers in the zone. Any Active Directory users you add to the EnterpriseUnixAdmins group and provision a UNIX profile for will be able to run any command with root-level permissions using their Active Directory credentials.

The Active Directory user you added to the EnterpriseUnixAdmins group can now log on and run privileged commands on the UNIX or Linux computers you are using for evaluation.

To verify the user can run privileged commands using Active Directory credentials

1. Log on to the Linux or UNIX computer using the Active Directory logon name and password you created for the UNIX administrator.
2. Open a terminal on the Linux or UNIX computer.
3. Run a command that requires root-level privileges.

    For examples, run the dzinfo command to view the rights and roles for the UNIX Login user you created Adding and provisioning an evaluation user and group.

    dzinfo user_name

    Because you are logged on as the Active Directory user and not invoking the command using your role assignment, the command displays an error message indicating that you are not allowed to view authorization information for another user.
4. Re-run the command using your role assignment by typing dzdo before the command.

dzdo dzinfo user_name

The command runs successfully and returns information about the evaluation user similar to this partial output.

User: lois.lane
Forced into restricted environment: No

Role Name Avail Restricted Env
--- --- ---
UNIXLogin/Headquarters Yes None
Effective rights:
Password login
Non password login
Allow normal shell

Audit level:
AuditIfPossible

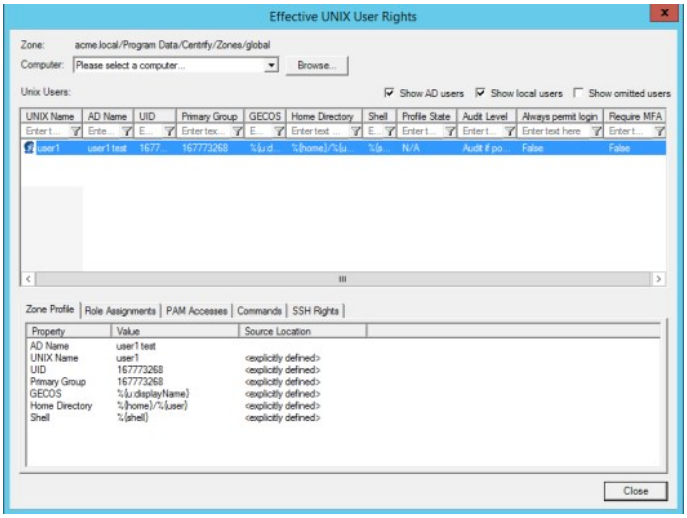
Viewing Effective Rights

Often, you need to see which users have what privileges in a zone. Access Manager provides you a single view of all of the effective users in a zone and lets you tab through their account properties.

To view effective rights for Linux and UNIX users

1. Open Access Manager.
2. Expand Zones, right-click your parent zone name, then select **Show Effective UNIX User Rights**.

   For example, the following illustrates the effective users in the evaluation zone.



3. Select a user, then click the tabs to see details about that user's profile, role assignments and UNIX rights.

Creating Child Zones and a Service Administrator Role

In many cases, you don't want a service administrator to have root privileges. For example, there's no reason to give database or web service administrators root-level privileges if their role only requires limited access to a few privileged operations.

To illustrate how to grant more limited privileges to an administrator, you will now create a role that gives an Apache server administrator permission a few specific tasks, such as edit the Apache configuration file and start and stop the Apache service. In this scenario, you will also create child zones to further limit the Apache administrator's authority to just the computers in the child zones.

To create child zones

1. Open Access Manager.
2. Expand Zones, right-click your parent zone name, then select **Create Child Zone.**
3. Type a Zone name (Nevada) and a brief description (Western field office), then click **Next**.
4. Click **Finish**.
5. Repeat Step 1 through Step 4 giving the second child zone a different name (Delaware) and description (Eastern web farm office).
6. Expand Child Zones and each new zone you created to view the nodes of the child zones.

To create a new Active Directory user and group for Apache administrators

1. Open Active Directory Users and Computers and create a new **User** object.
    1. Fill in the First, Last, and the User logon name fields.
    2. Type and confirm a password and select the Password never expires option.
    3. Acknowledge the warning, click Next, then click Finish.
2. Open Active Directory Users and Computers and create a new **Group** object in the UNIX Groups organizational unit.
    1. For the Group name, enter ApacheAdmins.
    2. Select Global as the scope for the group and Security for the type of group, then click OK.
3. Add the web administrator to the ApacheAdmins group.
    1. Select the user you created in Step 1, right-click and select Add to a group.
    2. Select the ApacheAdmins group, then click OK.
4. Provision a UNIX profile for the new user using Access Manager.
    1. Expand the Zones node and select the Headquarters, right-click, then select **Add User.**
    2. Select the user you created for web administration.
    3. Select Define user UNIX profile only and deselect Assign roles.
    4. Accept the default values for all profile properties.
    5. Review your selections, click Next, then click Finish.

Defining Command Rights and a New Role for Apache Administrators

You are now ready to create the privileged commands and role definition for the Apache administrators much as you did for the UNIX administrators. However, in this scenario, you will add the following new commands:

web_edit_http_config	vi /etc/httpd/conf	Edit the httpd daemon configuration file

web_apachectl	apachectl *:	Front end command for managing the httpd daemon
web_httpasswd	htpasswd *	Create and update HTTP server user name and password file

These commands will be added to a new role definition, ApacheAdminRights. As an alternative to creating the commands and role manually using Access Manager, as you did in the previous section, the following steps illustrate how you can use an ADEdit script.

ADEdit is a command-line scripting environment included with the Delinea Agent for *NIX. You can use ADEdit commands and scripts to modify Active Directory objects interactively directly from a UNIX or Linux computer terminal. The sample script ApacheAdminRole illustrates how you can use an ADEdit script to create UNIX rights and an Apache administrator role. This sample script is located in the /usr/share/centrifydc/samples/adedit directory on the UNIX or Linux computer where you have installed the Delinea Agent.

To create the ApacheAdmin commands and the ApacheAdminRights role

1. Log on to the Linux or UNIX computer using the Active Directory logon name and password you created for the UNIX administrator.
2. Open a terminal on the Linux or UNIX computer.
3. Change the directory to /usr/share/centrifydc/samples/adedit.
4. Run the ApacheAdminRole script.

   ./ApacheAdminRole

   If you see the error /bin/env: bad interpreter: No such file or directory, try changing the first line in the script to #!/usr/bin/env adedit.
5. Follow the prompts displayed to provide the following information for connecting to Active Directory:
   - Domain name.
   - The Active Directory account name that has administrator privileges in the organizational unit you're using for the Delinea zones.
   - The password for the Active Directory account.
6. Select the zone from the list of zones in your domain.

   For example, enter 2 to create the commands and role in the Nevada child zone or 3 to create the commands and role in the Delaware zone. The script then creates the commands and the role in the selected zone.

Verifying the Success of the Script

You can verify the new command rights and role in Access Manager.

To verify the script created command rights new role

1. Open Access Manager.
2. Expand the Nevada or Delaware child zone, then expand Role Definitions.
3. Select the ApacheAdminRights role to view the new command rights in the right pane.

   The new rights are also listed in the under the child zone UNIX Right Definitions > Commands node. If the new role is not listed, right-click, then select Refresh.

Adding Rights to the New Role Definition

The ApacheAdminRole script created the new UNIX command rights for Apache-related tasks. However, the Apache administrators require a few more rights to do their job. For example, the ApacheAdminRights role created using the sample script does not include the UNIX Login right for any computers.

To add more rights to the ApacheAdminRights role

1. Open Access Manager.
2. Expand the Nevada or Delaware child zone, then expand Role Definitions.
3. Select the ApacheAdminRights role, right-click, then select **Add Right**.
4. Select the Nevada or Delaware child zone from the list of zone to restrict the list of rights to the rights available in the child zone.
5. Select the following default rights:
    - login-all to allow Apache administrators to log on.
    - ssh to allow Apache administrators to use the PAM secure shell client application.
    - sshd to allow Apache administrators to use the secure shell server application.
    - dzssh-scp to allow Apache administrators to use the secure copy application.
    - dzssh-sftp to allow Apache administrators to use the secure file transfer application.
6. Click **OK**.

Assigning the Apache Administrator Role to a Group

You can now assign the ApacheAdminRights role to the Active Directory ApacheAdmins group. The members of this group will only have the Apache access rights on the computers in the Nevada or Delaware child zone you selected. Outside of the selected zone, members will have no access rights on any UNIX computers.

To assign the ApacheAdminRights role to the Apache administrators

1. Open Access Manager.
2. Expand the Nevada or Delaware child zone and its Authorization node.
3. Select Role Assignments, right-click, then select **Assign Role.**
4. Select the ApacheAdminRights role, then click **OK**.
5. Click **Add AD Account**.
    1. Change the object to Find from User to Group, then search for and select the ApacheAdmins group, then click OK.
    2. Click OK to complete the role assignment.

Deploying Group Policies to UNIX Computers

Centrify provides group policy templates for managing UNIX and Linux computers. The group policies are centrally managed through the Group Policy Management Editor, but modify configuration settings on the managed computers where they are applied. This mechanism allows you to manage the group policy settings from a single location and have them applied on remote UNIX and Linux computers.

To illustrate how to configure and apply group policies, you will create a Group Policy Object for the Centrify organizational unit.

To load and apply group policies for UNIX and Linux computers

1. Open the Group Policy Management utility (gpmc.msc) and expand your evaluation domain.
2. Right-click the Delinea organizational unit, and select **Create a GPO in this domain, and Link it here.**
3. Type a name for the new GPO (UNIX policies), then click OK.
4. Expand the Delinea organizational unit, right-click the GPO, then select **Edit**.
5. Expand the Computer Configuration > Policies node and select **Delinea Settings**.
6. Right-click and select **Add/Remove Templates**
7. Click **Add** and select all of the templates listed, click **Open**, then click **OK**.

   This step adds both computer and user group policies under the Delinea Settings node. Expand Delinea Settings to explore the specific policies available. You can click the Explain tab for any group policy to see more information about what it does. The remainder of this section illustrates how you would enable and configure a few simple policies for Delinea-managed. You should note that all policies—including Delinea group policies—are "Not configured" by default.

Configuring User Mapping by Group Policy

To illustrate how to configure a Delinea group policy, you will enable the Set user mapping policy. This policy maps a UNIX user, for example root, to an Active Directory user account, for example Amy.Adams. After this policy is set, root attempts to log on must use the mapped Active Directory user's credentials.

To configure a Delinea group policy

1. Expand Delinea Settings > DirectControl Settings, scroll down and double-click the **Set user mapping** policy.
2. Select **Enabled**, then click **Add**.
3. Type the UNIX user account name (root).
4. Click Browse to search for and select the Active Directory account to use, then click **OK**.
5. Click **OK** to enable the policy.

   Note: If you enable this policy, the root user in the zone will **not** be able to log in to the managed computers in the zone.

Configuring Password Prompts

There are several group policies that enable you to customize the text displayed when a user attempts to log on to a managed computer. For example, you can customize the text displayed when a password is expiring in a certain number of days or when authentication fails. To illustrate how to configure the Delinea group policies for password-related prompts, you will enable the Set login password prompt policy.

1. Expand **Delinea Settings > DirectControl Settings > Password Prompts** and double-click **Set login password prompt**.
2. Select **Enabled**.
3. Type the text string you want displayed, then click **OK**.

Next Steps

You now have a basic foundation for working with Server Suite software. You have created a parent zone and child zones, provisioned users to log on to computers in those zones, defined rights and roles in different zones, and granted Active Directory users and groups specific rights by assigning them to roles. You've also seen how to apply and configure group policies for Centrify-managed computers. From here, you can experiment on your own or explore some of the additional tools that Server Suite provides.

Exploring Additional Management Tools

In configuring a basic evaluation environment, you saw how you can use Active Directory to centrally manage user accounts, access privileges, and group policies on Linux and UNIX computers through Server Suite zones. This chapter introduces some of the additional Server Suite tools that you can use to manage the UNIX users and computers in your organization.

Adding UNIX Profiles Automatically

Adding UNIX user accounts to Active Directory on a large scale poses several challenges:

- Provisioning: How do you provision large numbers of UNIX users and map them to unique Active Directory user objects?
- Assigning roles: Once the UNIX users have profiles stored in Active Directory, how do you give each user just the privileges required?
- Accommodating legacy UIDs: How do you migrate UNIX users who have different UIDs on different servers and maintain existing file ownership requirements?

One strategy for adding and managing a large number of UNIX profiles is to use the Zone Provisioning Agent and provisioning properties. The Zone Provisioning Agent can automatically provision new users with the full complement of UNIX profile attributes when you add them to an Active Directory group. Configuring the environment to illustrate automated provisioning with the Zone Provisioning Agent, however, requires several steps that are only applicable if you choose that deployment scenario.

The following steps summarize the process, but are not recommended for an evaluation.

To deploy the Zone Provisioning Agent

1. Create an Active Directory service account with the "Log on as a service" user right.
2. Open the Centrify Zone Provisioning Agent Configuration Panel and configure the service to use the service account you created for it.
3. Create or identify the Active Directory groups you will use as source groups for UNIX users.
4. Set the provisioning properties for the zone or zones where users will be automatically provisioned.

   For example, open Access Manager, select the parent zone, right-click, then select Properties to see the Provisioning properties. You can then set the Active Directory source group and how you want UNIX attributes to be automatically generated.
5. Migrate all existing users using the appropriate override attributes into zones to preserve their profiles.
6. Start the Zone Provisioning Agent service.

Keep in mind that the Zone Provisioning Agent takes over all user provisioning if enabled for a zone. After you start the service, you cannot use the Access Manager **Add User** option to add a user to the zone. This ensures that all UIDs are unique in the domain.

If you configure the Zone Provisioning Agent, you can add and remove users from selected Active Directory groups to automatically add or remove their UNIX profiles in a zone.

To add users after configuring zone provisioning

1. Open the users.txt file in the /usr/share/centrifydc/samples/adedit directory to add more or change names.

   Use an editor that does not insert a carriage return at the end of each line. Each line must end with a line feed.
2. Run the AddUnixUsers sample script in the directory to create the Active Directory account for each UNIX user and add each user to the Active Directory UNIX Users group.

   ./AddUnixUsers users.txt.
3. Follow the prompts displayed to provide the following information for connecting to Active Directory:
   - Domain name.
   - The Active Directory account name that has administrator privileges in the organizational unit you're using for the Delinea zones.
   - The password for the Active Directory account.
4. Type an initial password that meets the Active Directory requirements to be used for all of the accounts added.
5. Open the Delinea Zone Provisioning Agent Configuration Panel and click **Restart**.
6. Open Access Manager or Active Directory Users and Computers and assign users to the appropriate Active Directory groups to assign rights.

Delinea

Managing UNIX Information from a UNIX Terminal

Many organizations find it least disruptive for their UNIX administrators to continue to manage their UNIX and Linux computers directly from their own computer rather than from a Windows computer. If you plan to manage zones, UNIX user and group accounts, access privileges, roles, and role assignments from a UNIX or Linux computer, you can use the command-line tools described in this section.

Using UNIX Commands

This following table summarizes the most commonly used Centrify command line programs.

adcheck	/usr/share/centrifydc/bin	Performs operating system, network, and Active Directory tests to verify a computer meets the system requirements for a successful installation. For example, the install.sh script runs the adcheck program.
adedit	/usr/bin	Starts the adedit application for interactive commands or running scripts For more information about the adedit application, see Using ADEdit.
adflush	/usr/sbin	Clears the computer's agent cache. Use this after you have made changes to Active Directory accounts to remove and replace the previous values.
adgpupdate	/usr/bin	Retrieves group policies from the Active Directory domain controller and applies the policy settings to the local computer and current user immediately. If you do not use the command, group policies are automatically updated at a random interval between 90 and 120 minutes.
adinfo	/usr/bin	Displays summary or detailed diagnostic information for the managed computer.
adjoin	/usr/sbin	Joins the local computer to an Active Directory domain, organizational unit and zone.
adleave	/usr/sbin	Removes the local computer from the Active Directory domain.
adpasswd	/usr/bin	Changes the Active Directory account password for the current user or a specified user.
adquery	/usr/bin	Queries Active Directory for information about users and groups.
dzinfo	/usr/bin	Displays information about the effective rights and roles for the current login account.
dzdo	/usr/bin	Enables you to run privileged commands as root or another user.

Some UNIX commands require you to be logged on as root or as a user with root privileges. Other commands allow different operations or return different results if you are logged on as root. For the complete list of Server Suite command line programs you can run on Linux and UNIX computers, see the *Administrator's Guide for Linux and UNIX*. For detailed information about the options available for any command, see the man page for that command.

Using ADEdit

The Server Suite Agent for *NIX also includes the Tcl-based ADEdit program. ADEdit has two basic components:

- the adedit command-line application
- the ade_lib Tcl library

ADEdit provides a scripting language that you can use to bind to one or more Active Directory domain controllers. You can then use ADEdit to retrieve, modify, create, and delete Active Directory objects of any kind, including Server Suite specific objects such as zones, rights, and roles. For example, you used ADEdit and a sample script to create rights and a role in Defining command rights and a new role for Apache administrators.

The following sections introduce a few of the key features for ADEdit. For more information about using ADEdit commands and the ade_lib library, see the *ADEdit Command Reference and Scripting Guide*.

ADEdit Application

ADEdit uses Tcl as its scripting language. The Tcl scripting language includes all standard programming features, such as variables, logical operators, and predefined functions (called "procedures" in Tcl). The ADEdit application also includes a Tcl interpreter and Tcl core commands, which allow it to execute standard Tcl scripts, and a comprehensive set of its own commands designed to manage Server Suite-specific objects in Active Directory.

You can use ADEdit to execute individual commands interactively or to execute sets of commands together in the form of an ADEdit script.

ade_lib Tcl Library

The ade_lib Tcl library is a collection of Tcl procedures that provide helper functions for common Centrify-specific management tasks such as listing zone information for a domain or creating an Active Directory user. You can include ade_lib in other ADEdit scripts to use its commands.

Using adedit Sample Scripts

The Server Suite Agent for *NIX includes several sample adedit scripts that you can run in your evaluation environment. The scripts are in the /usr/share/centrifydc/samples/adedit directory on the UNIX or Linux computer where you have the agent installed.

To run scripts that have the .sh extension, enter /bin/sh filename.sh.

To run scripts that do not have an extension, you can just enter ./filename.

> **Note**: If you get the error /bin/env: bad interpreter: No such file or directory when you run a script, this means that the env command is not in the /bin directory. In most cases, it is in /usr/bin instead. To fix this, change the first line in the script to:

```
#!/usr/bin/env adedit
```

The following table lists the sample scripts and the arguments.

AddUnixUsers	users.txt	none
ApacheAdminRole	none	none
computers-report	-domain domain_name -u AD_user_name -sep separator	-m -p password Use -m if you want to authenticate using the computer account credentials instead of an Active Directory user account. If using an Active Directory user account, use -p if you want to include the user's password in the command line. If you don't specify this option, you are prompted for the password.
CreateChildZones	-d domain_name -z parent_zone_name -u AD_user_name	-p password Use -p if you want to include the user's password in the command line. If you don't specify this option, you are prompted for the password.
CreateParentZone	-d domain_name -z zone_name	none
GetChildZones	none	none
GetComputers	none	none
GetGroups	none	none
getopt-example	-d domain_name -u AD_user_name	-p password Use -p if you want to include the user's password in the command line. If you don't specify this option, you are prompted for the password.
Getusers	none	none
GetZones	none	none
MakeRole	Role_apacheAdmin.txt	none

MktDept.sh	List of names, for example, Mary, Joe, and Lance	none
useracc-report	-domain domain_name -u AD_user_name -sep separator	-m -p password Use -m if you want to authenticate using the computer account credentials instead of an Active Directory user account. If using an Active Directory user account, use -p if you want to include the user's password in the command line. If you don't specify this option, you are prompted for the password.
user-report	-z zone_distinguished_name	-m -p password Use -m if you want to authenticate using the computer account credentials instead of an Active Directory user account. If using an Active Directory user account, use -p if you want to include the user's password in the command line. If you don't specify this option, you are prompted for the password.

For more information about the sample scripts and how they can be used or modified, see the *ADEdit Command Reference and Scripting Guide*.

Next Steps

You have now explored some of the additional tools available for working with Server Suite-managed computers, including the basic features of ADEdit sample scripts and default reports. You are now ready to see how you can use the audit and monitoring service to capture, replay, and manage user sessions on managed Linux and UNIX computers.

Auditing Sessions

This chapter describes how to install and use the Delinea Administration and Services components. The auditing service is a process on each managed UNIX and Linux computer that captures user session input and output and transfers this information to a collector service. The collector service forwards the audited sessions to a database, where it is available for review and replay.

Install Auditing Components on Windows

For the evaluation, you are going to install the auditing infrastructure on a single Windows computer. To complete these steps, you will install a Microsoft SQL Server database for the evaluation environment, a single collector, and the Audit Manager and Audit Analyzer consoles from the Audit & Monitoring Service setup program. You have already installed the auditing service on the Linux or UNIX computer you are using for the evaluation.

To install auditing components on the Windows computer

1. On the physical or virtual computer where you downloaded Server Suite software, double-click **autorun.**
2. On the Getting Started page, click **Audit & Monitor**.
3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement, click **I accept the terms in the license agreement**, then click **Next**.
5. Select both **Centrify Administration** and **Centrify Services** to install all components, then click **Next**.
6. Accept the default location for installing files by clicking **Next**, then click **Next** to proceed with the installation.
7. Confirm that the Launch Configuration Wizard box is selected by default, then click **Finish**.
8. Click **Exit** to close the **Getting Started** page.

Configure a New Audit Installation

An audit *installation* is a logical object similar to an Active Directory forest or site. It encompasses all of the auditing components you deploy—agents, collectors, audit stores, audit store databases, management database, and consoles—regardless of how they are distributed on your network. The installation also defines the scope of audit data available. All queries and reports are against the audit data contained within the logical boundary of the installation.

To create a new installation for auditing in the evaluation environment

1. If you have launched the new installation wizard automatically, at the **Welcome** page, click **Next**.

   You can also use Audit Manager to launch the new installation wizard.

2. In the New Installation wizard, accept the default audit installation name by clicking **Next**.

   For the evaluation, use the default installation name to automatically collect the sessions cached on the managed computers. If you use a different name, you must manually specify the installation an audited computer should use.

3. Select the option to create a new management database and verify the SQL Server computer name, instance name, and database name are correct, then click **Next**.

4. Select **Use the default NT AUTHORITY\SYSTEM account** to run the stored procedures that read and write information to the management database, then click **Next**.

5. Type the license key you received, then click **Add** or click **Import** to import the keys directly from a file, then click **Next**.

6. Accept the default location for publishing installation information, then click **Next**.

7. Select the installation-wide auditing options you want to enable, then click **Next**.

   For the evaluation, select **Enable video capture recording of user activity** to capture shell activity on the audited computer, then click **Next**. Do not select the options that disallow the review and deletion of your own sessions.

8. Review details about the installation and management database, then click **Next**.

   If you have SQL Server system administrator (sa) privileges and can connect to the SQL Server instance, the wizard automatically creates the management database.

9. Select the **Launch Add Audit Store Wizard** option if you want to start the Add Audit Store wizard, then click **Finish**.

Enabling Linux Desktop Auditing

In addition to shell auditing, for some Linux systems you can also enable desktop auditing. When desktop auditing is enabled, the user's entire screen is continuously monitored to record all graphical interactions. More specifically, desktop auditing captures the following:

- The application name and window title when the user switches the focus to that application. For example, if a user opens a web browser or a terminal window.
- Changes to the application window title that currently has focus. For example, if a user opens a web browser and goes to a new web page, desktop auditing records the title of a web page.

The supported platforms for Linux desktop auditing are as follows:

- RHEL 6, 7, and 8 with GNOME v3
- CentOS 6, 7, and 8 with GNOME v3

Linux sessions must be running X as the primary display manager (not Wayland).

Linux desktop auditing requires shell session auditing.

To enable desktop auditing on a Linux computer

1. Log on as a user with root privileges.
2. Run dacontrol with the -x option or the --desktop-audit option:

   ```
   dacontrol -x
   ```

   ```
   dacontrol --desktop-audit
   ```

   To enable both shell and desktop auditing at the same time, use both the -e and -x options:

   ```
   dacontrol -e -x
   ```

3. Run dainfo to verify that desktop auditing has been enabled.

   For example, the relevant information from the dainfo command looks like this:

```
Pinging adclient: adclient is available
 Daemon status: Online
 Current installation: 'DirectAudit' (configured locally)
 Current collector: test.acme.com:5063:HOST/test.acme.com@acme.com
 DirectAudit NSS module: Active
 ...
 DirectAudit desktop auditing: Enabled
 User (root) audited status: Yes
```

When you enable auditing, the desktop auditing module shows as Enabled. You can also see if auditing is enabled or not for a system in the Audit Manager console.

Verify that Auditing is Enabled

After the auditing infrastructure is installed and configured, you are ready to audit activity on the managed computers where the Server Suite Agent is installed.

To check that auditing is enabled on the managed computer

1. Log on to the managed computer as root.
2. Run the following command verify auditing is enabled:

   ```
   dacontrol -e
   ```

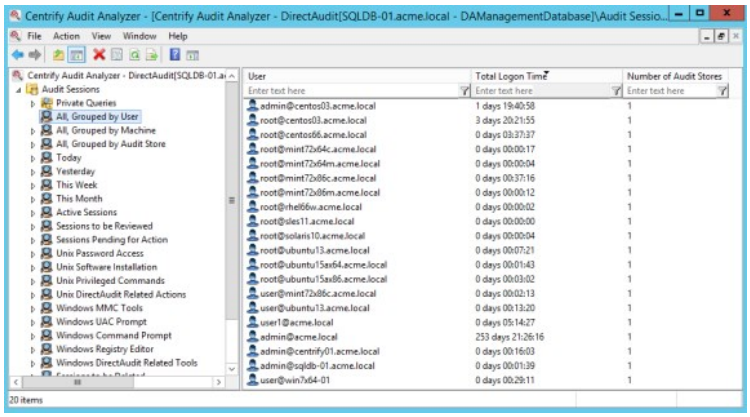
   This command will enable auditing or display a message indicated that auditing is already enabled.

Within a few minutes the collector service should start to retrieve session activity for the managed computer. For more information about configuring and managing the auditing infrastructure, see the Unexpected Link Text.

Viewing Sessions with Predefined Queries

After you have started collecting user activity on a managed computer, you can use Audit Analyzer to view and replay the sessions captured. For example, you can open Audit Analyzer and select **Active Sessions** to see sessions that are currently in progress.

Audit Analyzer includes many predefined queries like the Active Sessions query that you can use to find the sessions in which you are interested. To access the predefined queries, expand Audit Sessions. You can then select a predefined query to display a list of the audited sessions that meet the conditions of that query. For example, if you want to search for sessions by user, you can select the "All, Grouped by User" query, then select the specific user whose sessions are of interest to see a list of all the sessions captured for that user. For example, in the right pane, you would select a user from the list.



After you select a specific user, Audit Analyzer displays detailed information about each of that user's sessions. For each session, Audit Analyzer lists the user name who started the session, the user display name, the account name used during the session, the name of the audited computer, the audit store used, start and end time, current state, whether the audited session is a console or terminal client session, the review status of the session, the name of the user that modified the status, the size of the session in kilobytes, and any comments that have been added to the session.

In addition to the predefined queries for audited sessions, Audit Analyzer includes predefined queries for audit trail events and predefined queries for basic reports. You can explore these queries on your own as you capture additional activity.

Replaying a Session

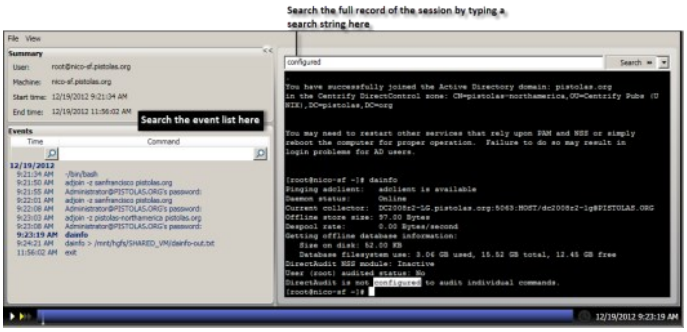
If you accepted the defaults when you created the installation for auditing, you should have video capture auditing enabled. Video capture auditing records all standard input (stdin), standard output (stdout), and standard error (stderr) activity that occurs on the managed computer. With video capture enabled, you can select a session, right-click, then select **Replay** to review the session in the session player.

At this point in the evaluation, you have had very limited activity on the Linux or UNIX computer you are managing and auditing. Before replaying any sessions, you might want to log on to the managed computer and run several simple UNIX shell commands, then close the UNIX terminal and log off.

To replay the sample session

1. Open Audit Analyzer from the desktop icon.
2. Click **Today** in the left pane to list the sessions that have run today.
3. Select the session that has UNIX shell command activity, right-click, then click **Replay** to display the session player.

   The left pane of the session player displays a summary of activity. You can search on any column to find events of interest. You can also search for a specific text string. For example:



4. Click the **Play**/**Pause** icon at the bottom of the session player to start or stop the session you are viewing.

   You can also fast forward session playback by clicking the **Speed control** icon to play back at 2x or 3x the normal speed. The dark blue playback line across the bottom of the window represents the total time of the session. You can drag the **Timepoint needle** to go directly to a specific point in the session.

   The **Real-time** icon toggles to allow you to play back a session as it was recorded in real time or move swiftly from one user action to the next. The **Session point** in the lower right corner identifies the date and time of the current point in the session playback.

5. Close the session player.

Managing Audited Sessions

You can right-click any session to view an indexed list of the commands captured, export the session activity to another format for sharing or further analysis, update the review status for the session, or delete the session.

Using Command Summaries

You can view a list of the commands the user executed in a selected session by right-clicking the session, then selecting Indexed Command List. This option provides a summary of user activity so that you can quickly scan for events of interest or for suspicious activity without replaying activity. You can then start the session player from a specific command in the list by selecting the command and clicking **Replay**.

Exporting Sessions

You can export session activity to several different formats to enable you to share information for review and analysis. After selecting a session, you can right-click to export the session to the following formats:

- As a plain text (TXT) file that includes the time of each input and output event that occurred during the session.
- As a comma separated values (CSV) file where each row represents a single command input or output line from the terminal window.
- As a Microsoft Windows Media Video (WMV) file can be played by using any media player that supports the WMV format. This option enables you to share the video capture of activity with auditors or other users who don't have access to Audit Analyzer. You should note, however, that WMV files do not include all of the information available in the session player. For example, exporting a session to a WMV files does not preserve information such as the session summary that includes the user name, computer, start and end time for the session and the summary of events.
- As a uniform resource identifier (URI) by selecting **Copy Session URI**. This option enables you to share the session with auditors or other users who don't have access to Audit Analyzer. Once copied to the clipboard, you can paste the URI into a browser to open the session for replay.

Viewing and Editing Session Properties

If you select a session, right-click, then select **Properties** you can view detailed information about the session, including the type of session, the session start and end times, the zone where the session took place, the audit store where the session is stored, details about the user whose activity was recorded and computer where the session ran, and the current status of the session. From the properties section, you can also view the current review status for the session, when the review status was last modified, and who made the change to the review status. You can also click on the Reviewers tab in Properties to see the list of users that are authorized to review the session, change the status of the session, and add comments to the session. By clicking the Comments tab, you can also view and add comments to the session. For example, you might want to use the Comments tab to add details about what to look for in a session to assist a reviewer or to provide additional information when you change the review status of a session.

Updating Review Status for a Session

You can use the **Update Review Status** for a session to distinguish sessions that warrant attention and to mark their progress through your review cycle. For example, if you find a session that warrants analysis, you might right-click to select Update Review Status, then select **To be Reviewed**. After you select a new status, you are prompted to add comments and the session is added to the appropriate predefined query in the left pane. For example, if you selected To be Reviewed status, the session to the **Sessions to be Reviewed** list.

After you review the session and you determine it needs further action, you might select the **Pending for Action** review status. Selecting this status removes the session from **Sessions to be Reviewed** list and adds it to **Sessions Pending for Action** list.

Deleting Sessions

You can select a session, right-click, then select **Delete** to delete a session after you have finished reviewing activity and taken appropriate action or when it is no longer needed. Selecting this option deletes the session from all predefined and custom query lists. For example, if you delete the session from the results for the **Today** predefined query, the session might also be deleted from the results for the predefined **Sessions to be Reviewed** query or any shared or private queries where it was previously listed.

Creating Custom Queries

In addition to the predefined queries, you can use Audit Analyzer to create your own queries for locating sessions using specific criteria. For example, you might want to find all sessions that contain the string sudo or that ran a specific program. To search for these sessions, you can create a custom query definition.

For audited sessions, you can create:

- Quick queries
- Private queries
- Shared queries

If you create a quick, private, or shared query, a new node is added to the Audit Analyzer console for that type of query under the Audit Sessions node. If you want to search for audit trail events, you can also create queries for audit events, which are added to Audit Analyzer under the Audit Events node.

To create a new custom query

1. Open Audit Analyzer, select Audit Sessions, right-click, then select one of the following options for a new query:
    - New Quick Query
    - New Private Query
    - New Shared Query
2. Type a name and description for the query.
3. Select the type of sessions that you want the query to find.

    For example, select UNIX sessions to limit the search to only include UNIX sessions. By default, new queries search for both UNIX and Windows sessions.
4. Select an attribute for grouping query results, if applicable.
5. Select an attribute for ordering query results within each group, if applicable.
6. Click **Add** to add search criteria to filter the results of the query.
7. Select an appropriate attribute from the Attribute list based on the sessions you want to find.
8. Select the appropriate criteria for the attribute you have selected, then click **OK**.

    The specific selections you can make depend on the attribute selected. For example, if the attribute is **Review status**, you can choose between "Equals" and "Not equals" and the specific review status you want to find., such as "To be Reviewed." If you select the attribute **Comment**, you can specify "Contains any of" and type the text string that you want to find any part of.
9. Click **Add** to add another filter to the criteria for the query, or click **OK** to save the query and find the sessions that match the criteria you have specified.

Frequently Asked Questions

This section provides answers to common questions and information about specific features that are not applicable for all organizations. You should review the questions covered to see if there are any topics of interest or are relevant to your situation.

How Do I Accommodate Legacy or Conflicting Identity Information?

If you plan to migrate existing UNIX and Linux users to Active Directory, you might have users that already have different login names or UIDs on multiple UNIX or Linux computers. For file and directory ownership to continue uninterrupted, those users must be able to continue using those legacy identity attributes.

To accommodate different login names and UIDs on different computers, you can create computer-level overrides that let you change just those UNIX attributes you need to change for individual UNIX or Linux computers. The legacy attributes remain tied to a single Active Directory account, but enable you to deploy with no changes to your existing environment.

To set computer-level overrides

1. Expand Zones and parent and child zones to find the zone for the computer requiring an override.
2. Expand **Computers** to display the computer requiring an override.
3. Expand the computer name and UNIX Data.
4. Right-click **Users** under the selected computer, click **Add User to Zone**
5. Search for and select the Active Directory user.
6. Select the UNIX properties to change in the user's UNIX profile.

   For example, you can change the UID used for the selected user. The new profile attribute is only used on the computer where you make the change.
7. Set the new value, then click **OK**.

For all other computers in the selected zone and in other zones, the user's UNIX profile remains unchanged. You can change any or all profile attributes on other computers to accommodate your legacy identity information.

Can I have Separate Role Assignments for Specific Computers?

Yes. Server Suite-managed computers get their role assignments from three places:

- Parent and child zone role assignments made in the Authorization node.
- Role assignments made at the computer level.
- Role assignments made in the zone's computer roles.

Generally, you start assigning roles at the child zone and then the computer role levels. However, there are occasions when you need to make the role assignment for a single computer. In this case, you use the computer-level override functionality.

To make a role assignment as a computer-level override

1. Expand Zones and parent and child zones to find the zone for the computer requiring an override.
2. Expand **Computers** to display the computer requiring an override.
3. Expand the computer name and select Role Assignments.
4. Right-click **Role Assignments** under the selected computer, click **Assign Role.**
5. Select the role requiring a computer-specific assignment.
6. Click **Add AD Account** to search for and select a user or group.

How Can I Manage Access Rules for Computers in Different Zones?

You can use computer roles—groups of computers with a common purpose—to simplify assigning access roles. A computer role is simply an Active Directory group of computers. You create this group because a specific set of computers have something in common. For example, you can create a security group for all Oracle database servers in your organization, or all Oracle servers in a specific location, or all Oracle servers owned by a certain team of administrators. The same computers might be in multiple Active Directory groups, but each group defines a specific purpose. The computers might also be in the same zone or different zones.

A computer role enables you to associate an Active Directory group of computers with a specific set of access rules that apply to just that set of computers.

To create a computer role that defines access rules for a group of computers

1. Create Active Directory groups for the sets of users who have specific access rights.

   For example, you might create a group for OracleUsers and a group for OracleAdmins in the Delinea UNIX Groups organizational unit.

2. In Access Manager, expand Zones and parent and child zones to find the zone for the computer requiring a computer role.

3. Expand Authorization, right-click Computer Roles, then select **Create Computer Role**.

4. Type the name and description, then select **Create group** to create the Active Directory security group for the computers than share a common purpose.

   For example, create a new global group named Oracle Servers.

5. In Access Manager, create or identify the access rights and role definitions that will be specifically applicable for the set of computers.

   For example, define the access rights appropriate for the Oracle users and for the Oracle administrators.

   Add role definitions for the Oracle users (OracleLoginRights) and administrators (OracleAdminRights), then add the appropriate rights to each role.

6. Assign the role definitions to the appropriate Active Directory groups.

   For example, assign the OracleLoginRights role to the OracleUsers group and the OracleAdminRights role to the OracleAdmins group.

7. Add the computers to the computer role group.

   For example, expand Computer Roles and Oracle servers, right-click Members, then select **Add Computer** to add each Oracle server to the Members node.

Delinea

How do I Manage Access Privileges during Application Development?

In-house application development and deployment typically require three sets of computers, each with its own set of users and privileges:

- Development: The set of computers with the source code and tools for application development. You only want your developers and maybe one or two users to have access to these computers.
- Test: The set of computers used by QA to confirm that the application conforms to specifications. You only want the QA staff to have access to these computers.
- Production: The computers deployed throughout the enterprise. You don't want developers or QA to have access to these computers.

You can use computer roles to ensure that only specified users have access at each stage. In this case, you would define two computer roles in the zone:

- DevelopmentSystems
- TestSystems

Then, you would do the following:

- Create Developer and Tester groups in Active Directory.
- Create Developer and Tester roles and add the rights in Access Manager.
- Assign the roles to the groups in the DevelopmentSystems and TestSystems roles.
- Add the development and test computers as a member to each role.

Now, only the members of the Developer and Tester Active Directory groups have access to the corresponding computer role's member computers.

How do I Terminate a User Account but Keep the Account Profile?

When a user leaves the company, you might want to retain their account profile to ensure all of the files they created on your organization's UNIX and Linux computers have an owner. You can use the predefined listed role to retain an account profile with no access privileges.

To create the group and assign the role

1. Create an Active Directory group in the UNIX Groups organizational unit called Listed. In the description enter, Terminated users.
2. In Access Manager, expand Zones and find the zone where the account profile is required.
3. Expand Authorization and Role Assignments, then select **Assign Role**.
4. Select the listed role, click **Add AD Account**, search for and select the select the Listed Active Directory group, then click **OK**.

To terminate a user

1. Remove the user account from all of the UNIX Groups that have access rights.
2. Verify that the user has no role assignments and no effective rights in any zone.
3. Add the user account to the Listed group.

If the user rejoins the company, you simply delete the user from the Listed group and add the user to groups, as needed.

Can Active Directory Credentials be used to Log in to Applications?

Yes. Delinea provides additional packages that let you configure single sign-on for Apache, Tomcat, JBoss, WebSphere and WebLogic web servers, and for Oracle, DB2, and SAP database applications.

Can Active Directory Credentials be used for Phone and Tablet Users?

Yes. Delinea offers software that enables you to authenticate users on iOS and Android devices before they can access their company email, web, and SaaS applications. A separate evaluation package is available for you to try out mobile device management for smart phones or tablets. Contact your sales representative for a free evaluation.

How Do I Migrate from NIS Maps to Server Suite?

Access Manager provides an extension that enables you to import and manage NIS network maps in Active Directory on a zone-by-zone basis. For UNIX and Linux computers and applications that submit lookup requests directly to a NIS server listening on the NIS port, you can also deploy the Delinea Network Information Service, adnisd, to receive and respond to NIS client requests from the NIS map information stored in Active Directory.

Removing Software after an Evaluation

The evaluation software can only be used for a limited time. After you complete the evaluation, you should remove the software to free up space on the physical or virtual computers you used for the evaluation. This section describes the steps for removing components from the Windows computer you used for the evaluation and the UNIX or Linux computer you added to Active Directory.

- Removing Authentication and Privilege Services
- Removing the Audit and Monitoring Service
- Removing Server Suite Agents

Removing Authentication and Privilege Services

The most efficient way to remove the Authentication & Privilege Services components from a Windows computer is to rerun the setup program that installed them.

To remove Authentication & Privilege components

1. On the physical or virtual computer where you downloaded Server Suite software, double-click **autorun.**
2. On the **Getting Started** page, click **Authentication & Privilege**.
3. At the **Welcome** page, click **Next**.
4. Select **Uninstall**, then click **Next**.
5. Review the list of software to be removed, then click **Next**.
6. Click **Finish** to exit the wizard.

The Authentication & Privilege components are now removed from the host Windows computer. You should note, however, that these steps do not remove any of the Active Directory organizational units, users, or groups you used for the evaluation. You should manually remove these objects with Active Directory Users and Computers or ADSI Edit.

Removing the Audit and Monitoring Service

The most efficient way to remove Delinea Management Services components from a Windows computer is to rerun the setup program that installed them.

To remove the Audit & Monitoring Service components

1. On the physical or virtual computer where you downloaded Server Suite software, double-click **autorun.**
2. On the **Getting Started** page, click **Audit & Monitor**.
3. Select **Uninstall**, then click **Next**.
4. Click **Finish** to exit the wizard.

The Audit & Monitoring Service components are now removed from the host Windows computer. You should note, however, that these steps do not remove the installation service connection point, databases, or database instances. You should manually remove these objects with ADSI Edit and Microsoft SQL Server Management Studio.

Removing Server Suite Agents

Follow the steps below to remove the Server Suite Agent for *NIX and command line programs—such as adinfo, adjoin, adquery, dacontrol, and dzinfo— from the computer.

You can rerun the **install.sh** script interactively or silently using a configuration file to remove Server Suite software from a managed computer.

To remove the agent and other packages using the install.sh script

1. Log on and open a terminal on the managed computer.
2. Run the adleave command to remove the computer from the domain controller.

   ```
   adleave -u administratorname
   ```

   The user name you specify with the **administratorname** argument should be an account with Active Directory administrator privileges.
3. Type the password for the an account name you specified.
4. Change to the directory that contains the extracted agent package.
5. Run the installation script.

   ```
   /bin/sh install.sh
   ```

   The script determines the Server Suite software you have installed on the computer and displays the details for you to review.
6. Enter **E** to proceed.
7. Confirm the removal of packages by entering **Y** to proceed.
8. Enter **Y** to reboot the computer after removing software packages.

Delinea Server Suite Free (formerly Centrify Express)

The following Server Suite Free docs are available:

- Delinea Server Suite Free Guide
- Delinea Server Suite Free Administrator's Guide for Linux and UNIX

In previous releases, this product was called Centrify Express.

The *Server Suite Free Administrator's Guide for Linux and UNIX* describes how to install, configure, and use the components in Delinea Server Suite Free for UNIX and Linux. Delinea Server Suite Free products are available for free to provide identity and access control for cross-platform data centers using Active Directory. With support for a wide range of operating systems, hypervisors, and applications, Delinea Server SuiteAgents can help your organization strengthen security and regulatory compliance while reducing IT expenses and costly interruptions to user productivity.

Delinea Server Suite Agents provide simplified cross-platform integration with Active Directory. In most cases, Delinea Server Suite Free agents require little or no configuration, and are available for download directly from the Delinea web site. By installing Delinea Server Suite Agents, you can add UNIX and Linux computers to Active Directory, authenticate user credentials from a central identity store, and support local and remote cross-platform single sign-on at no cost.

The following topics are available:

This chapter provides an introduction to Delinea Server Suite Free for Linux and UNIX, including a brief overview of how Delinea can help you take advantage of your investment in Active Directory.

Key Components

Delinea bundles products and features in different editions to address different customer requirements. The Delinea Server Suite Free family of products provides the most basic set of functionality and is available for free from the Delinea website.

The main Delinea components that enable cross-platform authentication and authorization services using Active Directory are platform-specific agents. Agents are packaged in compressed platform-specific files that you can download and extract to enable non-Windows computers to join an Active Directory domain. After you install an agent and join a domain, Active Directory users are authenticated on the UNIX or Linux computer without any further configuration.

The Delinea Server Suite Free family of products also includes Kerberos-enabled versions of OpenSSH and PuTTY packages.

Features Not Supported by Delinea Server Suite Free

Taken together, Delinea Server Suite Free products provide a solid foundation of functionality that is suitable for many organizations without upgrading to Server Suite. However, Delinea Server Suite Free does not provide central management of policies, delegated administration, identity control, role-based access rights, or auditing services.

If your organization outgrows the basic functionality of Delinea Server Suite Free, you can upgrade to Server Suite to take advantage of these additional features.

The following table describes features that are limited or not enabled in Delinea Server Suite Free.

Centralized identify and access management	You cannot centrally manage user and group profiles, control access privileges on specific computers, or delegate administrative activities.
Group policies	You cannot centrally manage configuration settings for non-Windows computers and users.
Auditing	You cannot audit user session activity.
Role-based authorization and access rights	You cannot define rights, roles, and role assignments to enforce role-based access to privileged commands and other operations.
Unlimited Delinea managed computers	The number of Delinea-managed computers that can be connected to the Active Directory domain at the same time is limited. The limit is described in the End User License Agreement (EULA) that is specific to Delinea Server Suite Free.
User login controls	You can only use a limited set of parameters to control which users or groups are granted or denied access.
Active Directory lookup filtering	You cannot use the NSS override parameters to filter Active Directory lookups requests.
The adcert command	You cannot use the adcert command, which enables certificate operations to be performed directly on agent-managed UNIX computers.
Data isolation and encryption	You cannot dynamically isolate and encrypt data in motion.

You must upgrade to a license version of Server Suite to use any of these features.

Managed Computers are Active Directory Clients

The agent enables non-Windows servers and workstations to participate in an Active Directory domain as Active Directory clients. You install the agent on each computer that you want to make part of an Active Directory domain. After you install the agent and join a domain on a computer, the computer is considered a Delinea managed computer. The agent then manages the connection to Active Directory domain controllers when users log on or connect to the computer remotely.

What the Agent Does

The agent makes a computer look and behave like a Windows client computer to Active Directory. The agent performs the following key tasks:

- Joins the computer to an Active Directory domain.
- Communicates with Active Directory to authenticate users when they log on.
- Caches users credentials for offline access.
- Enforces Active Directory authentication and password policies.
- Provides a Kerberos environment so that existing Kerberos applications work transparently with Active Directory.

Agents Consist of Multiple Components

Agents provide an integrated set of services that enable programs and applications to use Active Directory. The core agent service is the adclient process. The adclient process handles all of the direct communication with Active Directory and coordinates with other services to process requests for authentication, authorization, directory assistance, or policy updates.

Other services handle specific types of operations. For example, the pam_centrifydc module enables any PAM-enabled program, such as ftpd, telnetd, login, and sshd, to authenticate using Active Directory. A custom NSS module modifies the nsswitch.conf configuration file so that system look-up requests use the information in Active Directory. A configurable local cache stores user credentials and other information for offline access and network efficiency.

In addition to the core agent services, agents can include Delinea compiled versions of other programs, such as OpenSSH and OpenLDAP, to work with Active Directory.

Provisioning is Automatic

When you deploy an agent on a computer, the agent adds the computer account to Active Directory and automatically creates consistent UIDs across the joined domain for Active Directory users with access to the computer. The agent authenticates all valid Active Directory users without any configuration or account management. Because there is only one zone for the forest, you can deploy without creating any zones of your own. Because profiles are generated automatically, you do not need to configure any zone properties or manage who has access to which subsets of UNIX and Linux computers.

Deciding Whether to Use Zones

The primary reason to use Delinea Server Suite Free is that it enables Active Directory authentication without any planning, manual configuration, or account management. A primary limitation to using Delinea Server Suite Free is that all computers are placed in a single, automatically defined zone.

Zones provide a powerful and flexible structure for managing user identities, role-based access controls, and delegated administrative authority. However, deciding on the best strategy for using zones requires some planning and preparation. If your organization does not require more than one zone, you can begin deploying agents immediately.

Working With a Single zone

Delinea Server Suite Free is designed for organizations that do not want to centrally manage user profiles, role assignments, or administrative activities. After the agent is installed, all valid Active Directory users and groups in the entire Active Directory forest are automatically assigned a unique UNIX profile that allows them to log on. Because the Delinea Server Suite Free agent requires no configuration or central management, it is most suitable for organizations that:

- Want to add computers to a domain quickly without configuring any zones.
- Do not need to maintain or manage existing UIDs and GIDs.
- Have a limited number of users and domains.
- Have a relatively flat organizational structure.

If a single zone suits the needs of your organization, Delinea Server Suite Free provides a no-cost, cross-platform solution for authentication services. If your organization grows in size and complexity or if you want more granular access controls, you can upgrade to a licensed version of Delinea software at a later time. For more information about Delinea service offerings and Server Suite, see "Comparing Delinea Server Suite Free to other services" on page 30.

All Active Directory Users Have Access

After you install an agent and join an Active Directory domain, all of the users and groups in the Active Directory forest automatically become valid users and groups for the joined computer. In addition, all Active Directory users defined in any forest with a two-way trust relationship with the forest of the joined domain are valid users for the joined computer.

> **Note**: If a computer joins a domain and the domain has a one-way trust relationship with another domain, users and groups in the trusted domain **do not** become valid users and groups on the computer.

By default, all valid users can perform the following tasks:

- Log on interactively to the shell or a desktop program and use standard programs such as telnet, ssh, and ftp.
- Log on to a computer that is disconnected from the network or unable to access Active Directory, if they have successfully logged on and been authenticated by Active Directory previously.
- Manage their Active Directory passwords directly from the command line, provided they can connect to Active Directory.

How the Agent Generates Profile Attributes

Computers with a Delinea Server Suite Free agent always connect to the domain through the Auto Zone. In the Auto Zone, user profile attributes, such as the UID, default shell, and home directory are automatically derived from user attributes in Active Directory or from configuration parameters. No local account information is used or migrated into Active Directory.

When an Active Directory user logs on to a UNIX or Linux computer for the first time, the agent automatically creates a 31-bit UID for the user and a 31-bit GID for any groups to which the user belongs. To create unique GIDs and UIDs, the agent creates a prefix from the last 9 bits of the user or group Security Identifier and combines it with the lower 22 bits of the user or group relative identifier (RID).

Although the agent caches these UID and GID values, they are not stored in Active Directory. You cannot edit or change them in any way with Active Directory Users and Computers (ADUC). If the cache expires, the agent uses the same algorithm to create the same UID and GID the next time the user logs on so you are guaranteed consistent ownership for files and resources. In addition, users who log on to more than one computer will have the same generated UID on each managed computer.

> **Note**: All profile attributes—including the UID and GID values—are stored in Active Directory. If you upgrade to a licensed version of Delinea software, you can migrate and manipulate UID and GID properties for individual computers. You can also map multiple UIDs to a single Active Directory account to allow different UIDs settings on different computers for the same user account. This type of manipulation is not possible when using Auto Zone and Delinea Server Suite Free agents.

In addition to the UID and GID, the agent automatically creates a home directory for the user with all the associated profile and configuration files. The location for the home directory is:

- UNIX or Linux: /home/username
- Mac OS X: /Users/username

Deploying an agent does not affect local users. User accounts that are defined in the local /etc/passwd directory can still log on. If you want to control access through Active Directory, however, you should create Active Directory accounts for each user. After you verify user access for the Active Directory user, you can then either delete the local account, or map the local users on each computer to an Active Directory account to preserve access to current home directories and files. For more information about mapping accounts, see "Mapping local accounts to Active Directory" on page 56.

Using Delinea Server Suite Free to Deploy Agents

With Delinea Server Suite Free, you can discover and analyze computers on your network or in the cloud, then download and install or update the correct agent for each discovered computer. You can also use Delinea Server Suite Free to manage account information for remote UNIX users and groups, and run programs on the computers discovered.

Like other Delinea products, you can download Delinea Server Suite Free agents from the Delinea website.

Comparing Delinea Server Suite Free to other services

Delinea Server Suite Free provides a subset of the features available in authentication and privilege elevation services. Over time, this basic set of functionality may be insufficient. Depending on the needs of your organization, you may want to upgrade the Server Suite you use to take advantage of additional feature sets. The following table provides a brief description of the services available.

Delinea Server Suite Free	Free software that provides basic integration with Active Directory for authenticating users.
Server Suite	Commercial offering that provides a full complement of services to ensure the security of your infrastructure and prevent the breaches that can result when privileged accounts are compromised. With Server Suite, you can protect your organization in a variety of ways. For example, you can:: Require users to log in as themselves. Enforce least-privilege access for administrators and end-users. Control shared access to privileged accounts. Audit and monitor user activity and what takes place during privileged sessions. Isolate and encrypt sensitive information transmitted over the network.

This section provides step-by-step instructions for installing the Delinea Agent on a computer and joining the computer to the Active Directory domain.

Selecting a Deployment Option

The agent must be installed on each computer you want to manage. You must also specify an Active Directory domain for the agent to join either during the installation process or after the agent files are installed.

You can install and manage agent packages independently by running an installation script, package management program, or software distribution tool locally or remotely on individual computers.

For more information, see Options for Deploying Agent Packages.

Installing and Using Delinea Server Suite Free

Delinea Server Suite Free provides a Windows-based MMC console and a self-contained database that stores information about the computers and accounts discovered on the network or in the cloud.

Minimum Hardware Requirements

You can install Server Suite Free on a single Windows computer with a 64bit operating system.

In general, Delinea recommends the following minimum hardware configuration:

- 2 GB RAM
- 1 GB free disc space
- 2 GHz processor

Network Connectivity Requirements

To download and deploy software, you must have network connectivity or an Internet connection between the Windows computer where Access Manager is installed and the computers where you want to deploy the agent. Delinea recommends that you install Access Manager on a computer that allows outbound Internet connections and connectivity between the Windows computer and each computer you want to manage.

Account Credential Requirements

To install software on remote computers and join Active Directory domains, you must have access to an account with appropriate permissions:

- To run privileged commands, you should have access to the root account, the local Administrator account, or an account that has been granted escalated privileges using su or sudo and settings in a sudoers configuration file.
- To join a domain, you need an Active Directory account and password that has permission to add computers to the domain.

Depending on your organization, the Active Directory account might be required to be a member of the Domain Admins group. If you are not sure whether you have permission to add computers to the domain using your own Active Directory account, check with the Active Directory administrator for your site.

Download the Software and Run the Setup Program

If you have a computer that meets the requirements and the appropriate account information, you can download Server Suite Free.

To download and install Delinea Server Suite Free:

1. Go to the Delinea website and register an account, if you have not previously registered
2. Click the **Download** link.
3. Open the downloaded file to start the setup program.
4. Follow the prompts displayed to accept the license agreement and select a location for program files.
5. Install the agents on the desired computers. For details, see "Options for deploying agent packages" on page 37.

Options for Deploying Agent Packages

You can download individual Delinea Agent packages for the platforms you support and install the software in one of the following ways:

- Run the installation script (install-express.sh) locally on any computer and respond to the prompts displayed.
- Create a configuration file and run the installation script remotely on any computer in silent mode.
- Use the install or update operations in the native package installer for your operating environment.

If you want to use one of these installation options and need more information, see the appropriate section.

Install Interactively on a Computer

You must install a platform-specific agent on each computer you want to manage through Active Directory.

The installation script automatically checks the operating system, disk space, DNS resolution, network connectivity, and other requirements on a target computer before installing. You can run this script interactively on any supported UNIX, Linux, or Mac computer and respond to the prompts displayed.

To install agent packages on a computer interactively:

1. Go to the Delinea website and download the Delinea Server Suite Free agent for the platform you want to support.
2. Select the file you downloaded and unzip and extract the contents using the appropriate operating system commands. For example:

   ```
   gunzip -d centrify-package-platform-arch.tgz
   ```

   ```
   tar -xf centrify-package-platform-arch.tar
   ```

3. Run the install-express.sh script to start the installation on the local computer. For example:

   ```
   ./install-express.sh
   ```

4. Follow the prompts displayed to check the computer for potential issues, install the agent, and join a domain automatically at the conclusion of the installation.

   If the adcheck program finds potential issues, you might see warning or error messages. Depending on the issue reported, you might have to make changes to the computer before continuing or after installation.

   For most prompts, you can accept the default by pressing Enter. When prompted for the Active Directory domain, type the fully qualified name of the Active Directory domain to join.

   You must also type the user name and password for an Active Directory user with permission to add computers to the domain.

5. After you have responded to all of the prompts displayed, review your selections, and then enter **Y** to continue with the installation and reboot the computer.

Using Other Programs to Install

If you want to manually install a software package using a native installation program instead of the installation script, use the installation commands and options that are appropriate for the local operating environment. For example, if your operating system supports a package installer, such as Red Hat Package Manager (rpm), SMIT or YAST programs, you can use any of those programs to install the agent.

> **Note**: Delinea recommends that you use the installation script to automatically check a computer for issues and join the computer to a domain.

To install an agent using a native installation program

1. Log on as or switch to the root user.
2. If the software package is a compressed file, unzip and extract the contents. For example, on Red Hat Linux:

   ```
   gunzip -d centrify-*-rhel5-x86_64.tgz
   ```

   ```
   tar -xf centrify-*-rhel5-x86_64.tar
   ```

3. Run the appropriate command for installing the package based on the local computer's operating system or package manager you want to use. For example, on Red Hat Linux:

   ```
   rpm -Uvh centrifydc-*-rhel5-x86_64.rpm
   ```

4. Disable licensed features by running the adlicense --express command:

```
adlicense --express
```

Note: You must run the adlicense command to set the agent to run in Express mode. Express mode is used for the Server Suite Free product.

5. Join the domain by running the adjoin --workstation command, which connects you to Auto Zone:

```
adjoin --workstation domainName
```

Note: If you do not specify the --workstation option, the join operation will fail because adjoin will attempt to connect you to a specific zone rather than Auto Zone.

Verifying the Installation

When a computer is joined to Active Directory, all Active Directory users and groups defined for the forest, as well as any users defined in a two-way trusted forest, are valid users or groups for the joined computer. Therefore, after running the agent and joining the computer to a domain, you can log on as any Active Directory user.

1. Log on using an Active Directory user account.

   When a user logs in for the first time, the agent automatically creates a home directory for the new user.

2. Run the adinfo command to see information about the Active Directory configuration for the local computer. You should see output similar to the following:

```
Local host name: QA1
Joined to domain: sales.acme.com
Joined as: QA1.sales.acme.com
Pre-win2K name: QA1
Current DC: acme-dc1.sales.acme.com
Preferred site: Default-First-Site
Zone: Auto Zone
Last password set: 2014-04-01 12:01:31 PST
CentrifyDC mode: connected
Licensed Features: Disabled
```

   Note that licensed features are disabled and that the zone is Auto Zone. Creating actual zones requires a licensed copy of Delinea software.

Troubleshooting adcheck Errors

You can run adcheck before, during, or after installation to verify that your computer is configured properly. This utility performs three sets of checks that are controlled by the following options:

- -t os checks the operating system, disk size, and Perl and Samba installations.
- -t net checks DNS to verify that the local computer is configured correctly and that the DNS server is available and healthy.
- -t ad includes the -t net checks and verifies that the domain has a valid domain controller.

Correcting Errors for the Operating System Check

The -t os option performs a series of checks that verify operating-system basics for the computer on which you are installing the agent. If your computer fails one of these checks, upgrade the computer with a new operating system version, required patch, a new Perl or Samba version, or free up sufficient disk space.

Correcting Warnings and Errors for the Network Check

The -t net option performs a series of checks that verify that DNS is correctly configured on your local computer and that the DNS server is running properly. There is also a check to verify that you are running a supported version of OpenSSH.

Note: A supported version of OpenSSH is not automatically installed. You must choose to install it during a custom installation.

Because the agent uses DNS to locate the domain controllers for the Active Directory forest, the appropriate DNS nameservers need to be specified in the local /etc/resolv.conf file on each computer before the computer can join the domain. If you receive errors or warnings from these checks, you need to correct them before joining a domain. Each warning or error message provides some help to resolve the problem.

Correcting errors for the domain controller check

The -t ad option locates each domain controller in DNS and then does a port scan and DNS lookup of each. The checks for this option also verify the global catalog and verify clock and domain synchronization.

If you receive errors or warnings from these checks, you need to correct them before joining a domain. Each warning or error message provides some help to resolve the problem.

Joining a Domain After Installation

When you install the agent using installexpress.sh, you can automatically join that computer to an Active Directory domain. If you do not join the domain when you run the installation script, or if you leave a domain and want to rejoin, you can manually join a domain by using the adjoin command.

To manually join a domain, you must use the --workstation option to connect to Auto Zone.

To join an Active Directory domain manually on a Linux or UNIX computer:

1. Log in as or switch to the root user.
2. Run adjoin to join an existing Active Directory domain. You should join the domain using a fully-qualified domain name. You must specify the --workstation option.

   For example, to join the sales.acme.com domain with the user account dylan:

   ```
   adjoin --user dylan --workstation sales.acme.com
   ```

   The user account you specify must have permission to add computers to the specified domain. In some organizations, this account must be a member of the Domain Admins group. In other organizations, the account simply needs to be a valid domain user account. If you don't specify a user with the --user option, the Administrator account is used by default.
3. Type the password for the specified user account.

If the agent can connect to Active Directory and join the domain, a confirmation message is displayed. All Active Directory users and groups defined for the forest, as well as any users defined in a two-way trusted forest are valid users or groups for the joined computer.

Restarting Services

You may need to restart some services on computers where you have installed the agent so that those services will reread the name switch configuration file. For example, if you typically log on to the computer through a graphical desktop manager such as gdm, you need to either restart the gdm service or reboot the workstation to force the service to read the updated configuration before Active Directory users can log on.

The most common services that need to be restarted are sshd and gdm. If you are using these services, you should restart them. For example, to restart sshd:

```
/etc/init.d/sshd restart
```

As an alternative to restarting individual services, you can reboot the system to restart all services.

> **Note**: Because the applications and services on different servers may vary, Delinea recommends you reboot each computer to ensure all of the applications and services on the system read the configuration changes at your earliest convenience.

Upgrading Delinea Server Suite Free

To take advantage of features that are part of Server Suite—- for example to define roles that control access rights and apply group policies to computers and users—you must upgrade from Delinea Server Suite Free to a licensed copy of Server Suite. Upgrading to a licensed version of the product is a three-stage process that involves:

- Installing and upgrading components on Windows.
- Upgrading the agent to enable licensed features on managed UNIX and Linux computers.
- Adding optional packages that are not included in Delinea Server Suite Free.

Upgrading Windows Components

If you are upgrading to a licensed version of Server Suite, there are several additional components available for you to install depending on the services you want to deploy. For example, there are console extensions that enable you to edit group policies and manage NIS maps through Active Directory.

To install and upgrade licensed components on Windows:

1. Obtain a license key and media for Delinea Management Services.

   You can also download an evaluation copy directly from the Delinea website, but you must have a license key to use the software for more than a limited period of time.

2. On a Windows computer that is joined to the Active Directory domain, connect to the distribution media.

   If you received the software on a CD, the Getting Started page is displayed automatically or when you double-click the autorun.exe program.

3. Click Authentication & Privilege to start the setup program for authentication and privilege elevation components.

4. Follow the prompts displayed to accept the license agreement, select the components to install, and a location for files.

5. When setup is complete for the selected packages, click **Finish** to close the setup program.

Upgrading Agents on Managed Computers

To upgrade agents to a licensed product, you must run a command line program to enable licensed features on each managed computer.

To enable licensed features on managed computers:

1. Log on to the computer that is running a Delinea Server Suite Free agent.

2. Run the following command to search the Active Directory forest for the license key and to enable licensed features.

   ```
   adlicense --licensed
   ```

3. Run the following command to verify that licensing has been enabled:

   ```
   adinfo

   Local host name: qa1
   Joined to domain: acme.com
   Joined as: qa1.acme.com
   Pre-win2K name: qa1
   Current DC: acme-dc1.acme.com
   Preferred site: Default-First-Site
   Zone: Auto Zone
   Last password set: 2014-04-01 12:01:31 PST
   CentrifyDC mode: connected
   Licensed Features: Enabled
   ```

   Note: After enabling licensed features, the computer is still connected to Auto Zone. If you are not using zones to migrate existing user populations or define role-based access controls, you can leave the computer in Auto Zone. If you want to take advantage of zones, you must:

   - Create at least one zone using Access Manager, adedit, or another tool.
   - Run adleave to leave the Active Directory domain and Auto Zone.
   - Run adjoin to rejoin the Active Directory domain and a specified zone.

For information about creating and managing zones, using group policies, and other features, see the *Planning and Deployment Guide* and the *Administrator's Guide for Windows*.

Adding Optional Packages After Installation

Depending on the services you choose to deploy, there are several optional packages that might be available for you to use. To add these packages, you must rerun the installation script and select which packages to install.

To add optional packages on computers where the agent is installed:

1. Change to the appropriate directory on the CD or to the directory where you have copied or downloaded the agent package.

2. Run the standard installation script for the agent and follow the prompts displayed:

   ```
   ./install.sh
   ```

3. When you are prompted whether to keep, erase, or reinstall the currently installed packages:

- Accept the default (**K**, keep) for the currently installed packages.
- Type Y (**Y**, yes) for each package you want to add.

4. Follow the prompts displayed to set installation options, such as the option to run adcheck and reboot the computer after installation.

   The computer remains joined to the domain you previously joined, your existing /etc/centrifydc/centrifydc.conf file is backed up, and any modifications you have made to the file are migrated to the new version of the file.

5. Restart running services, such as login, sshd, or gdm, or reboot the computer to ensure all services use the updated configuration.

Removing Delinea Server Suite Free

On most managed computers, you can remove the agent and related files by running the uninstall.sh script. The uninstall.sh script is installed by default in the /usr/share/centrifydc/bin directory on each managed computer.

To remove the agent on a managed computer:

1. Log on to the computer where the agent is installed.
2. Run the uninstall.sh script. For example:

   ```
   /bin/sh /usr/share/centrifydc/bin/uninstall.sh
   ```

   The uninstall.sh script will detect whether the agent is currently installed on the local computer and will ask you whether you want to uninstall your current installation.

3. To uninstall, enter Y when prompted.

If you cannot locate or are unable to run the uninstall.sh script, you can use the appropriate command for the local package manager or operating environment to remove the agent and related files.

This chapter explains how to perform common administrative and end-user tasks on managed computers that have the Delinea Agent installed.

Logging on to Your Computer

You log on to a joined computer in the same way you log on locally. For example, you type a user name and password to start a console session, remote shell session, or a desktop manager. In most cases, you do not have to specify the domain name when you log on. However, you do need to type the Active Directory password for your account and the password must conform to the password policies defined for the domain.

You can use any of the following formats for the user name when you log on:

- Active Directory samAccountName or Mac OS X short name (jcool)
- Active Directory userPrincipalName (jcool@acme.com)
- Windows NTLM format for domain and user name (acme.comjcool)

You can also use any of these formats to locate users in Active Directory.

By default, the Delinea Agent uses the Active Directory samAccountName attribute or the Mac OS X short name for the UNIX profile user name. You can specify a different form for the UNIX user name by setting the value of the auto.schema.name.format parameter in the /etc/centrifydc/centrifydc.conf configuration file.

Getting Configuration Information

After you log on to a computer, you can use the adinfo command to see information about the Active Directory configuration for the local computer. For example, type `adinfo` to display a summary similar to the following:

```
Local host name: QA1
 Joined to domain: sales.acme.com
 Joined as: QA1.sales.acme.com
 Pre-win2K name: QA1
 Current DC: acme-dc1.sales.acme.com
 Preferred site: Default-First-Site
 Zone: Auto Zone
 Last password set: 2014-04-01 12:01:31 PST
 CentrifyDCmode: connected
 Licensed Features: Disabled
```

For Delinea Server Suite Free, licensed features are disabled and the only zone supported is Auto Zone. If you upgrade at a later time, the licensed features will be enabled, and you will be able to use zones to provide secure, granular access control and delegated administration for computers joined to a domain.

Applying Password Policies

The agent enforces all of the password policies you have defined in Active Directory for all valid user accounts in the forest. For example, if your policy requires that new users must change their password the next time they log on, they are prompted to change the password at the next log-on whether they use a Windows or UNIX computer.

The agent also checks passwords to make sure that they conform to Active Directory policies for length and complexity. If a new or changed password meets all of the criteria, the account is updated with the new information in Active Directory and the user logs on successfully.

If you have defined additional policies, such as a maximum duration, reuse policy, failed attempt and account lock out policy, workstation restrictions, and logon hour restrictions, the agent also enforces those policies. Like Windows, the agent displays a warning message each time a user logs on if the user's password is set to expire in a given number of days.

Changing Passwords

As an administrator, you can set, reset, or change the password for other users using Active Directory or from the UNIX command line. Individual users can also change their own password at any time using the adpasswd command.

Changing Your Own Password

If you attempt to log on but your password has expired, you are prompted to provide your old password, a new password, and to confirm your new password. You can also change your own password at any time using adpasswd.

To change your own password

1. At the UNIX command line, run the following command:

   adpasswd

2. Type your old password. When changing your own password, you must always provide your old password.
3. Type the new password. The password should conform to Active Directory password policies.
4. Retype the new password.

For more information about using adpasswd, see the adpasswd man page.

Changing Another User's Password

You can use the adpasswd command to change the password of another Active Directory user if you provide the user name and password of an administrative account with the authority to change another user's password.

To change the password for another user

1. At the UNIX command line, run the adpasswd command and specify an Active Directory administrative account name with the authority to change the password for users in the domain. For example, to use the admin user account to change the password for the user jane in the sales.acme.com domain:

   adpasswd --adminuser admin@acme.com jane@sales.acme.com

2. Type the password for the administrative account. For example:

   Administrator password: xxx

3. Type the new password for the user specified. Because you are changing another user's password, you are not prompted for an old password. For example:

   New password:

4. Retype the new password.

   Repeat password:

For more information about using adpasswd, see the adpasswd man page.

Working in Disconnected Mode

After an Active Directory user logs on to a computer successfully, the authentication is cached on the local computer. These credentials can then be used to authenticate the user in subsequent log on attempts if the user is disconnected from the network or if an Active Directory domain controller is not available.

If there are changes to an account while the account is running in disconnected mode, the changes do not take effect until the user reconnects to Active Directory to start a new session or access a new service. For example, if a user account is disabled or has its password changed in Active Directory while the user is disconnected from the network, the user can still log on and use the old password until reconnected to the network. After the user reconnects to Active Directory, the changes take effect and the user is denied access or prompted to provide an updated password. Because changing the password for an Active Directory account requires a connection to an Active Directory domain controller, users cannot change their own Active Directory password when working in disconnected mode.

> **Note**: If users log out of a session while disconnected from Active Directory, they can be authenticated using the information in the cache when they log back on because they have been successfully authenticated in a previous session. They cannot, however, be authenticated automatically to any additional services after logging back on. To enable automatic authentication for additional services, the user's credentials must be presented to the Key Distribution Center (KDC) then issued a ticket that can be presented to other services for unprompted, single sign-on authentication. Because the KDC is unavailable when disconnected from Active Directory, single sign-on authentication is also unavailable.

You can configure many aspects of how credentials are handled, including how frequently they are updated or discarded, through parameter settings in the centrifydc.conf configuration file. To configure how credentials are handled using group policies, you must upgrade to a licensed version of Delinea software.

Mapping Local Accounts to Active Directory

By default, local user accounts are valid on the computers that join the Active Directory domain. In some cases, you may want to manually map a local user account to an Active Directory account instead of using a generated profile. Mapping a local user account to an Active Directory account gives you Active Directory-based control over password policies, such as password length, complexity, and expiration period.

> **Note**: Mac OS X users can always log on using their local account password. Therefore, you cannot enforce Active Directory password policies for local Mac OS X user accounts.

Mapping local accounts to Active Directory is especially useful if you want to preserve access to a user's current home directory and files. For example, if a local user has a UID of 518 but the Delinea Agent generates a different UID for the user's profile, that user will not have file ownership permissions for his home directory and files.

To map a local account to an Active Directory account, you can set the pam.mapuser.username configuration parameter on any individual local computer. To configure account mapping using group policies, you must upgrade to a licensed version of Delinea software.

Using the pam.mapuser Parameter

To map a local user account to an Active Directory user by modifying the local centrifydc.confconfiguration file:

1. Create the Active Directory user account to use.

   On your Windows Active Directory computer, open Active Directory Users and Computers (ADUC). Navigate to the Users node, right click and select **New > User**.

   You should create a user logon name with the same name as the local user.

2. On the computer with the local account, open the centrifydc.conf configuration file.
3. Locate the pam.mapuser.*username* configuration parameter and un-comment the line to change the default setting.
4. Modify the local account mapping to identify the local user account you want mapped to the Active Directory user you created. For example:

   ```
   pam.mapuser.__joe.cool__: __joe.cool__
   ```

5. Save the changes to the configuration file, then run the `adreload` command to reload the configuration file and have the changes take effect.

Setting a Local Override Account

In most cases, every computer should have at least one account that can be authenticated locally to ensure that you can access the system when the network or Active Directory is not available or adclient is not running. By default, the local override account is set to the root user so that even if you map the root account to an Active Directory account, you can always log on locally using root@localhost and the local root account password.

You can change the default root override account or add additional local users by modifying the computer's centrifydc.conf configuration file. To configure a local override account using group policies, you must upgrade to a licensed version of Delinea software.

Using native telnet, ssh, and ftp programs

By default, authorized users can use standard programs and services such as telnet, ssh, and ftp. For telnet and ftp, you can use the packages installed with the operating system. For ssh operations, however, Delinea recommends that you install the Delinea-compiled version of OpenSSH instead of using the package provided with the operating system. You can download a free copy of OpenSSH from the Delinea website.

Using Samba

Delinea Server Suite Free supports the adbindproxy package, which contains the components to enable an open-source Samba file server to use the Delinea Agent and Active Directory to handle identity management and user credentials.

For more information, see the *Samba Integration Guide*.

Setting Auto Zone Configuration Parameters

Delinea Server Suite Free Agents support a set of configuration parameters specifically intended for computers that are connected to a domain through Auto Zone.

Because Auto Zone is a single zone for an entire forest, you can encounter problems such as UID and GID conflicts and slow searches. If you encounter these

problems, you may need to modify the default configuration. For information about how to set specific parameters to resolve UID and GID conflicts or improve search performance, see [Customizing Operations Using Configuration Parameters] (#customizing).

This chapter describes how to use diagnostic tools and log files to retrieve information about the operation of Delinea Agents and provides tips to help you identify and correct problems on managed computers.

Addressing Log On Failures

In most cases, valid Active Directory users should be able to log on to computers where you have deployed the agent without any configuration. If an attempt to log on fails, the problem is typically caused by one of the following:

- Users attempting to log on to a computer they are not authorized to use.
- Users do not have a valid Active Directory user account in the appropriate forest.
- Users have typed their non-Active Directory password or typed the wrong password more times than allowed.

If users report that they cannot access computer resources they think they should have access to, take the following steps to troubleshoot the problem:

1. Verify that the user has an Active Directory user account in the forest or in a forest with a two-way trust relationship.
2. Check that the account is not disabled or locked out because of repeated log-on failures.
3. Verify that there is an Active Directory domain controller available and that the computer a user is unable to log on to can connect to it and open a communication channel.

   For example, log on to the UNIX computer using a locally authenticated user, and run the ping command with the name of a domain controller in the forest. If the command receives a reply from the domain controller, the DNS service is functioning and the local computer is able to locate the domain controller on the network.

   If the ping command does not generate a reply, check your DNS configuration and check whether the local computer or the domain controller is disconnected from the network.
4. Use adinfo or Active Directory Users and Computers to check that the computer is joined to the domain.
5. Use adinfo to check whether the agent is currently running or disconnected.

   If the adinfo command reports the mode is disconnected, try restarting adclient and testing network response time. On a slow network, adclient may drop the connection to Active Directory if there is a long delay in response time.

   If the adinfo displays an error, try running adleave to leave Active Directory, re-run the adjoin command to re-join the domain. If a problem still exists, check the DNS host name of the local computer and the domain controller, the user name joining the domain, and the domain name you are using.
6. Check the clock synchronization between the local computer and the Active Directory domain controller.

   If the clocks are not synchronized, reset the system clock on the managed computer using the date command.
7. Check the contents of the system log files or the centrifydc.log file after the user attempts to log on. You can use information in this file to help determine whether the issue is with the configuration of the software or with the user's account.
8. Check for conflicts between local user accounts and the user profile generated by the agent.

If these steps do not reveal the problem, you can enable detailed logging of adclient activity using the addebug command. You can use the information in the /var/log/centrifydc.log file to further diagnose the problem or to provide information to Delinea Support.

Understanding Diagnostic Tools and Log Files

The agent includes some basic diagnostic tools and a comprehensive logging mechanism to help you trace the source of problems if they occur. These diagnostic tools and log files allow you to periodically check your environment and view information about agent operation, Active Directory connections, and the configuration settings for individual computers you manage.

Logging is not enabled by default for performance reasons. Once enabled, however, log files provide a detailed record of agent activity. This information can be used to analyze the behavior of adclient and communication with Active Directory to locate points of failure. However, log files and other diagnostic tools provide an internal view of operation and can be difficult to interpret. The log files are primarily intended for Delinea Support and technical staff.

In most cases, you should only enable logging when you need to troubleshoot unexpected behavior, authentication failures, or problems with connecting to

Active Directory or when requested to do so by Delinea Support. Other troubleshooting tools, such as command line programs, can be used at any time to collect or display information about your environment.

Configuring Logging

By default, the agent logs errors, warnings and informational messages in the syslog and /var/log/messages files along with other kernel and program messages. Although these files contain valuable information for tracking system operations and troubleshooting issues, occasionally you may find it useful to activate Delinea-specific logging and record that information in a log file.

Enabling Logging for the Agent

To enable logging on the agent:

1. Log in as or switch to the root user.
2. Run the addebug command:

   /usr/share/centrifydc/bin/addebug on

   Note: You must type the full path to the command because addebug is not included in the path by default.

   After you run this command, all of the agent activity is written to the /var/log/centrifydc.logfile. If the adclient process stops running while you have logging on, the addebug program records messages from PAM and NSS requests in the /var/centrifydc/centrify_client.log file. Therefore, you should also check that file location if you enable logging.

For performance and security reasons, you should only enable logging when necessary. For example, if you open a case with Delinea Support, the Support representative may request that you enable logging and submit log files to investigate your case. You should also limit logging to short periods of time while you or Delinea Support attempt to diagnose a problem. You should keep in mind that sensitive information may be written to this file and you should evaluate the contents of the file before giving others access to it.

When you are ready to stop logging activity, run the addebug off command.

Setting the Logging Level

You can define the level of detail written to the log by setting the log configuration parameter in the centrifydc.conf configuration file:

log: level

With this parameter, the log level works as a filter to define the type of information you are interested in and ensure that only the messages that meet the criteria are written to the log. For example, if you want to see warning and error messages but not informational messages, you can change the log level from INFO to WARN. By changing the log level, you can reduce the number of messages included in the log and record only messages that indicate a problem. Conversely, if you want to see more detail about system activity, you can change the log level to INFO or DEBUG to log information about operations that do not generate any warnings or errors.

You can use the following keywords to specify the type of information you want to record in the log file:

FATAL	Fatal error messages that indicate a system failure or other severe, critical event. In addition to being recorded in the system log, this type of message is typically written to the user's console. With this setting, only the most severe problems generate log file messages.
ERROR	System error messages for problems that may require operator intervention or from which system recovery is not likely. With this setting, both fatal and less-severe error events generate log file messages.
WARN	Warning messages that indicate an undesirable condition or describe a problem from which system recovery is likely. With this setting, warnings, errors, and fatal events generate log file messages.
INFO	Informational messages that describe operational status or provide event notification.

Logging Details for a Specific Component

By default, when you specify a logging level, it applies to all of the agent components that log activity. The logging system, however, provides a hierarchical organization of logical log names for the components within the agent and each of these logical logs can be configured to provide more targeted analysis of it specific operations. For example, if you set your base logging level to only report serious errors but you want to see informational, warning, and error messages for adclient, you can add a separate logging level parameter for the log messages generated by adclient:

```
# Use the following setting to set the base level of detail
# for logging to record Error messages:
 log: ERROR

# Add the name of the adclient logical log and specify the
# logging level to use for it and its children:
 log.com.centrify.adclient: INFO
```

Logging to the Circular In-memory Buffer

If the adclient process is interrupted or stops unexpectedly, a separate watchdog process (cdcwatch) automatically enables an in-memory circular buffer that writes log messages passed to the logging subsystem to help identify what operation the adclient process was performing when the problem occurred. The in-memory buffer is also mapped to an actual file, so that if there is a system crash or a core dump, the last messages leading up to the event are saved. Messages from the in-memory circular buffer have the prefix _cbuf, so they can be extracted from a core file using the strings command.

The in-memory circular buffer allows debug-level information to be automatically written to a log file even if debugging is turned off. It can be manually enabled by restarting the adclient process with the -M command line option. The default size of the buffer is 128K, which should be sufficient to log approximately 500 messages. Because enabling the buffer can impact performance, you should not manually enable the circular buffer or modify its size or logging level unless you are instructed to make the changes by Delinea Support.

Collecting Diagnostic Information

You can use the adinfo command to display or collect detailed diagnostic and configuration information for a local computer. Options control the type of information and level of detail displayed or collected. The options you are most likely to use to collect diagnostic information are the --config, --diag, or --support options, which require you to be logged in as root. You can redirect the output from any adinfo command to a file for further analysis or to forward information to Delinea Support.

For more information about the options available and the information returned with each option, see the adinfo man page.

To display the basic configuration information for the local computer, you can type:

```
adinfo
```

If the computer has joined a domain, this command displays information similar to the following:

```
Local host name: magnolia
 Joined to domain: ajax.org
 Joined as: magnolia.ajax.org
 Current DC: ginger.ajax.org
 Preferred site: Default-First-Site-Name
 Zone: Auto Zone
 Last password set: 2014-04-01 14:47:57 PST
 CentrifyDC mode: connected
 Licensed Features Disabled
```

Resolving Domain Name Service (DNS) issues

In some cases, you may encounter problems with authentication, authorization, or lookup requests because of your DNS configuration. The most common scenarios are:

- The Windows DNS server role is not configured to dynamically update service locator (SRV) records. These records enable Active Directory to find the nearest domain controller, Key Distribution Center (KDC), and Global Catalog (GC) for the site.
- The DNS servers do not publish the SRV records for the domain controllers that provide Active Directory service to the enterprise. These records must be available for computers to connect to Active Directory and locate required services.
- The DNS servers for the enterprise run on UNIX servers that are not configured to locate Active Directory domain controllers. In many cases, DNS servers for an enterprise are configured with a different domain namespace than Active Directory or Active Directory domain controllers are considered internal servers and not registered in the enterprise DNS.

If you encounter problems, you should contact your Active Directory administrator to determine whether the DNS server role is being used and if it is

configured to allow dynamic updates. If the Active Directory DNS server role is not being used to provide DNS to the enterprise, you should contact the DNS administrator to resolve the issue.

There are several possible scenarios:

- If the enterprise uses UNIX-based DNS servers instead of Active Directory-based DNS servers and DHCP, computers should have a snameserver entry in /etc/resolv.conf file that points to a valid DNS server.
- Forward and reverse lookup zones should be configured to allow enterprise DNS servers to locate Active Directory domain controllers.
- If the Active Directory domain namespace is different from the namespace registered in enterprise DNS servers, you should use the --name and --alias join option to resolve the namespace differences.
- If the enterprise DNS servers do not include records for Active Directory domain controllers, you can manually set the location of the Active Directory domain controller using parameters in the centrifydc.conf configuration file.

Command-line programs allow you to perform basic Active Directory administrative tasks directly from a UNIX shell or using a shell script. These commands use the underlying agent service library to enable you to perform administrative tasks, such as adding computers to an Active Directory domain, leaving the Active Directory domain, changing Active Directory passwords, and returning detailed Active Directory, network, and diagnostic information for a host computer.

Understanding When to use Command-Line Programs

Command-line programs are installed by default when you install the agent on a computer. Depending on the operating system, the commands are typically installed in one of the following directories:

```
/usr/sbin
/usr/bin
/usr/share/centrifydc/bin
```

In general, you should only use command-line programs when you must take action directly on a local computer. For example, if you want to join or leave a domain or set a new password while logged on to a shell, you may want to run a command interactively from that shell. You can also use command-line programs in scripts to perform administrative tasks programmatically.

Supported Command-Line Programs

Delinea Server Suite Free supports the following command-line programs:

adcache	The adcache program enables you to manually clear the local cache on a computer or check a cache file for a specific key value.
adcheck	The adcheck program verifies whether a local computer meets the system requirements for joining an Active Directory domain. This command checks whether the computer has sufficient disk and memory, a supported operating system and patch level, required libraries, and network connectivity to an Active Directory domain.
adclient	The adclient program manages most agent operations, and is normally started automatically when a computer starts up. In most cases, you should only run adclient directly from the command line if Delinea Support recommends you do so.
addebug	The addebug program starts or stops logging activity for agent operations.
addns	The addns program enables you to dynamically update DNS records on an Active Directory-based DNS server in environments where the DHCP server cannot update DNS records automatically.
adedit	The adedit program enables you to manage Active Directory and the agent through command-line commands and scripts.
adfinddomain	The adfinddomain program displays the domain controller associated with the Active Directory domain you specify.
adfixid	The adfixid program resolves UID and GID conflicts and enables you to change the ownership of a local user's files to match the user and group IDs defined for the user in Active Directory.
adflush	The adflush program clears the cache on a local computer.
adid	The adid program displays the real and effective UIDs and GIDs for the current user or a specified user.
adinfo	The adinfo program displays summary or detailed diagnostic and configuration information for a computer and its Active Directory domain.
adjoin	The adjoin program adds a computer to an Active Directory domain. This command configures a local computer to use Active Directory. No changes are made to authentication services or configuration files on a computer until you run the adjoin command. This command requires you to be logged on as root.
adkeytab	The adkeytab program enables you to create and manage Kerberos key tables (*.keytab files) and coordinate changes with the Kerberos key distribution center (KDC) provided by Active Directory.

adleave	The adleave program enables you to remove a computer from its current Active Directory domain or from the Active Directory forest entirely.
adlicense	The adlicense program enables or disables licensed features on a local computer. This command requires you to be logged on as root.
adpasswd	The adpasswd program changes the Active Directory account password for a user from within a UNIX shell.
adquery	The adquery program enables you to query Active Directory for information about users and groups from the command line on an agent-managed computer.
adreload	The adreload program forces the adclient process to reload configuration properties in the /etc/centrifydc.conf file and in other files in the /etc/centrifydc directory.
adrmlocal	The adrmlocal program reports and removes local user names that duplicate Active Directory user names.

Other commands that support Delinea operations are also installed in the directory with the commands shown in the preceding list, but they are not applicable to Delinea Server Suite Free agents.

Displaying Usage Information and Man Pages

To display a summary of usage information for a command-line program, type the command and the --help or -h option. For example, to see usage information for the adleave command, type:

```
adleave --help
```

The usage information includes a list of options and arguments, and a brief description of each option.

For more complete information about any command, you can review the information in the command's manual (man) page. For example, to see the manual page for the adleave command, type:

```
man adleave
```

In most organizations, the default settings in the /etc/centrifydc/centrifydc.conf configuration file are appropriate and do not require any customization. In some cases, however, you may find it useful to modify the default settings to optimize operations for your environment.

This section provides reference information for the configuration parameters that control the operations on managed computers. Parameters are also documented in comments within the centrifydc.conf file.

Auto Zone Configuration Parameters

auto.schema.primary.gid	Specifies the primary GID to use in the profiles automatically generated for users. To use this parameter: You should identify an existing group, such as Domain Users, to use as the primary group. You should verify that the auto.schema.private.group parameter is set to false. The default values for this parameter are platform-dependent, for example, 20 on Mac OS X computers and 65534 on Linux, HP-UX, Solaris, and AIX computers.
auto.schema.private.group	Specifies whether the agent should create dynamic private groups. If you set this parameter to true, the primary GID is set to the user's UID and a group is automatically created with a single member. The default value is false, enabling you to set the primary GID using the auto.schema.primary.gid parameter.
auto.schema.shell	Specifies the default shell for the logged in user. The default value is /bin/bash on Delinea Server Suite Free for Linux and UNIX and Linux and /bin/sh on other platforms, including Solaris, HP-UX, and AIX.
auto.schema.homedir	Specifies the home directory for logged in users. The default, if you do not specify this parameter, is: Mac OS X: /Users/%. Linux, HP-UX, and AIX: /home/% Solaris: /export/home/% The variable % is substituted at runtime and replaced with the logon name of the user who is logging on. For example, if the user jsmith logs on to a Delinea Server Suite Free for Linux and UNIX computer, the default home directory is set to: /Users/jsmith For example: auto.schema.homedir:/allusers/home/% This parameter is not used if the parameter auto.schema.use.adhomedir is set to true and a home directory is defined in Active Directory for the user. If auto.schema.use.adhomedir is false or no home directory is defined for the user in Active Directory, the home directory is set to the value defined for this parameter.
auto.schema.use.adhomedir	Specifies whether or not to use the Active Directory value for the home directory on Delinea Server Suite Free for Linux and UNIX computers. Set this parameter value to true to use the home directory defined in Active Directory. If you set this parameter to true but do not define a home directory in Active Directory, the value for auto.schema.homedir is used. Set this parameter to false if you do not want to use the home directory defined in Active Directory.
auto.schema.remote.file.service	Specifies the type of remote file service to use for mounting a network home directory on Mac OS X computers. The valid options are: SMB AFP For example: auto.schema.remote.file.service: SMB On Mac OS X computers, mounting a network directory requires that you specify the remote file service type. By identifying the remote file-service type using this parameter, you can type the network path in the format required by Active Directory: /server/share/path The agent then converts the Active Directory path into the format required by Mac OS X.
auto.schema.name.format	Specifies how Active Directory user names are transformed into UNIX login names. The valid options are: Active Directory samAccountName or Mac OS X short name (jcool) Active Directory userPrincipalName (jcool@acme.com) Windows NTLM format for domain and user name (acme.comjcool)
auto.schema.domain.prefix.*domain*	Specifies a unique prefix for a trusted domain. You must specify a whole number in the range of 0 - 511. The agent combines the prefix with the lower 22 bits of each user or group RID (relative identifier) to create unique UNIX user identifier (UID) and group identifier (GID) for each user and group. In most cases, this parameter is not necessary because the agent automatically generates the domain prefix from the user or group Security Identifier (SID). However, in a forest with a large number of domains or with cross-forest trusts, domain prefix conflicts are possible. If you attempt to join a computer to a domain and the agent detects conflicting domain prefixes, the join fails with a warning message. You can then set a unique prefix for the conflicting domains. To set this parameter, append the domain name and specify a prefix in the range 0 - 511. For example: auto.schema.domain.prefix.acme.com: 3 auto.schema.domain.prefix.finance.com: 4 auto.schema.domain.prefix.corp.com: 5

auto.schema.search.return.max	Specifies the maximum number of users to returned in search results. Because Auto Zone enables access to all users in a domain, a search could potentially return tens of thousands of users. This parameter causes the search to truncate after the specified number of users. The default is 1000 entries.
auto.schema.name.lower	Converts all user names and home directory names to lower case in Active Directory. Set to true to convert user names and home directory names to lowercase. Set to false to leave user names and home directories in their original upper, lower, or mixed case. The default for a new installation is true. The default for an upgrade installation is false.
auto.schema.iterate.cache	Specifies that user and group iteration take place only over cached users and groups. The valid options are: true restricts iteration to cached users and groups. false iterates over all users and groups. The default value is false.
adclient.ntlm.separators	Specifies the separators that can be used between the domain name and the user name when NTLM format is used. For example: adclient.ntlm.separators: +/ The default allows the following formats for the user joe in the acme.com domain: acme.com+joe acme.com/joe acme.comjoe **Note:** The backslash character () can be problematic on some UNIX shells, in which case you may need to specify domain user. The first character in the list is the one that adclient uses when generating NTLM names.

DNS-related configuration parameters

If computers cannot find the Active Directory domain controller, you can use parameters in the centrifydc.conf configuration file to manually identify the domain controllers and the Global Catalog server. You can also use configuration parameters to control how the DNS client processes DNS requests.

dns.dc.*domain_name*	Specifies one or more domain controllers to contact. You must specify the name of the domain controller, not its IP address. In addition, the domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller if you are not using DNS or if the DNS server cannot locate your domain controllers. For example, to manually specify the domain controller dc1.mylab.test in the mylab.test domain, you would add the following to the /etc/centrifydc/centrifydc.conf file: dns.dc.mylab.test: dc1.mylab.test To specify multiple servers for a domain, use a space to separate the domain controller server names. For example: dns.dc.mylab.test: dc1.mylab.test dc2.mylab.test The agent will attempt to connect to the domain controllers in the order specified.
dns.gc.*domain_name*	Specifies the domain controller that hosts the Global Catalog for a domain. If the Global Catalog is on a different domain controller than the domain controllers you specify with the dns.dc.*domain_name* parameter, you can use this parameter to specify the location of the Global Catalog. For example: dns.gc.mylab.test: dc3.mylab.test
dns.alive.resweep.interval	Controls how frequently the DNS client checks whether there is a faster DNS server available. The default interval for this check is one hour.
dns.sweep.pattern	Specifies the protocol and response time to use when the DNS client scans the network for available DNS servers. The dns.tcp.timeout and dns.udp.timeout parameters determine the amount of time to wait if the current server does not respond to a request. If the current server does not respond to a request within the specified time out period, it is considered down and the agent looks for a different server. If the DNS subsystem cannot find a live server, DNS is considered down, and the agent waits for the period of the dns.dead.resweep.interval parameter before performing a sweep to find a new server.
dns.tcp.timeout	Specifies the amount of time to wait if the current server does not respond to a TCP request. If the current server does not respond to a request within the specified time out period, it is considered down and the agent looks for a different server.
dns.udp.timeout	Specifies the amount of time to wait if the current server does not respond to a UDP request. If the current server does not respond to a request within the specified time out period, it is considered down and the agent looks for a different server.
dns.dead.resweep.interval	Specifies the amount of time to wait if DNS is before performing a sweep to find a new DNS server to use.

Delinea Server Suite Free Quick Start

Delinea Server Suite Free for Linux and UNIX is software that provides a basic set of features for cross-platform authentication and single sign-on using Active Directory. Delinea Server Suite Free for Linux and UNIX is available free of charge from the Delinea website.

After you install the agent and add the computer to an Active Directory domain, the computer becomes a Delinea managed computer and is treated as an Active Directory client.

What you should know about using the Delinea Server Suite Free agent:

- You don't need to install any software on any Windows computers or domain controllers.
- You don't need to make any configuration or schema changes in Active Directory.
- When users log on for the first time using an Active Directory user name and password, the agent automatically provisions the user identity, primary group ID, home directory, and default shell.
- Users can log on with a user name in the format that they are most comfortable with. For example, they can log on using a familiar UNIX user name or the same name they use to log on to a Windows computer.
- Local users with the same account name as Active Directory users are allowed to log on locally.
- Local users can be mapped to Active Directory users so that existing accounts can be migrated.

To use Delinea Server Suite Free, you need to install a Delinea Agent and join a domain. There are several options for installing the agent, including running an installation script locally, downloading a package from the Delinea repository to a local package manager, or installing remotely from a Windows computer using Delinea Server Suite Free.

The instruction here describe how to install using the default installation script. For information about installing using Delinea Server Suite Free, see Server Suite Free Administrator's Guide for Linux and UNIX.

For most prompts, you can accept the default by pressing Enter. When prompted for the Active Directory domain, type the fully qualified name of the Active Directory domain to join.

.

To install an agent on a local computer:

1. Make sure that you have the following information:
    - The root user name and password (or sudo rights) for the computer on which you are installing the agent.
    - The fully qualified name of the Active Directory domain to join.
    - The user name and password of an Active Directory account that has permission to join a computer to Active Directory.
2. Using an account with root permissions, log on to the target UNIX or Linux computer.
3. Go to the Delinea website and download the Delinea agent for the platform you want to support.
4. Select the file you downloaded and unzip and extract the contents using the appropriate operating system commands. For example:

    ```
    gunzip -d centrify-infrastructure-services-*<release>*-platform-arch.tgz
    ```

    ```
    tar -xf centrify-infrastructure-services-*<release>*-platform-arch.tar
    ```

5. Run the install-express.sh script to start the installation on the local computer. For example:

    ```
    ./install-express.sh
    ```

6. Follow the prompts displayed to check the computer for potential issues, install the agent, and join a domain automatically at the conclusion of the installation.

    If the adcheck program finds potential issues, you might see warning or error messages. Depending on the issue reported, you might have to make changes to the computer before continuing or after installation. For details about these messages and corrective actions, see Server Suite Free Administrator's Guide for Linux and UNIX.

    For most prompts, you can accept the default by pressing Enter. When prompted for the Active Directory domain, type the fully qualified name of the Active Directory domain to join.

You must also type the user name and password for an Active Directory user with permission to add computers to the domain.

7. After you have responded to all of the prompts displayed, review your selections, then enter **Y** to finish the installation and reboot the computer.

Steps that you can take after installing the agent include the following:

- **Explore Delinea Agent features.** For details about Delinea Server Suite Free features, see the *Server Suite Free Administrator's Guide for Linux and UNIX*.
- **Browse the Delinea community forum.** Go to http://community.centrify.com for discussions and articles about Delinea Server Suite Free for Linux and UNIX.
- **Install Delinea Server Suite Free.** If you would like to use a single console to manage the authentication and single sign-on services provided by Delinea Agents, download and install Delinea Server Suite Free. You can use Delinea Server Suite Free to discover and analyze computers on your network, download software and install or update the Delinea Agent on discovered computers. You can also use Delinea Server Suite Free to manage account information for remote UNIX users and groups, and to run programs on the computers that are discovered.
- **Upgrade to Server Suite.** Server Suite is available for purchase and includes a common set of authentication and authorization features, including:
  - **Zones.** Zones provide a powerful and flexible structure for managing user identities, role-based access controls, and delegated administrative authority. The ability to create and manage zones is a key element of the Server Suite family of products.
  - **Access controls.** PAM and auto schema parameters are provided to grant or deny access for specific users or groups.
  - **Policy support.** Policy-based enforcement of computer and user configuration settings.
  - **NIS support.** Support for NIS map integration and migration.
  - **Reports.** Standard out-of-the-box reports and a report creation wizard.
  - **Rights and roles.** Rights and role-based entitlements for user accounts and privileged commands.
  - **Commands and configuration parameters.** Advanced command line programs and configuration parameters for tuning operations.
  - **Smart card support.** For Mac OS X and Red Hat users, the ability to use PIV or CAC smart cards for authentication and single sign-on.

In addition to these common features, edition-specific features such as auditing, computer isolation and encryption, and application-specific services are available depending on which edition you purchase.

For more information about Server Suite, see the *Server Suite Free Administrator's Guide for Linux and UNIX*.

Integrations

The following Server Suite integrations docs are available:

- Authentication Guide fo IBM DB2
- Centrify-enabled PuTTY User's Guide
- RSA SecureID Token Config for UNIX/Linux computers
- Samba Integration Guide

In DB2, user and group authentication is performed by a facility that is external to the DB2 database management system, such as the operating system, a domain controller, or a Kerberos security system. It is accomplished using dynamically loadable libraries called security plug-ins.

The default IBM DB2 username/password plug-in authenticates users only in an NIS domain or in the /etc/passwd password file. If another security plug-in has not been explicitly configured, the user credentials provided in the connection request are authenticated by the security facility on the DB2 Universal Database (UDB) server. That is, the default plug-in sends the user ID and password to the operating system for validation.

Contents

Next Step:

DB2

Authorization is the process of determining access information about specific database objects and actions based on a supplied user ID. Privileges can be granted to specific users or to groups of users. Users that are a member of a group automatically inherit the group's privileges. As mentioned before, these users and groups are defined outside the DB2 UDB; for example, in Active Directory.

DB2 supports replacement plug-ins for authentication and authorization. The authentication plug-ins can replace the default user name and password method, and support alternative authentication methods including GSSAPI. DB2 also supports the use of multiple plug-ins for authentication.

Authentication for DB2 Security and Authentication Plug-Ins

Authentication Service for IBM DB2 package provides plug-ins that allow you to connect or attach to a DB2 database using either an Active Directory or a UNIX user identity. In addition, the package includes a group plug-in used for authorization.

The package provides two security plug-ins for authentication:

- centrifydc_db2userpass: a username/password plug-in to replace the DB2 default.
- centrifydc_db2gsskrb5: a GSSAPI plug-in for single sign on support.

The security plug-ins can be used independently or in conjunction with one another.

- If you specify and configure both the username/password plug-in and the GSSAPI plug-in, the GSSAPI plug-in is used when the user connects without specifying a user name and password. The user account can be on an Active Directory domain controller or UNIX computer. If the user does specify a user name and password, the username/password plug-in is used instead.
- If only the GSSAPI plug-in is configured, only Active Directory users can connect to the database instance. In addition, the Active Directory user name instead of the UNIX user name must be used in the SQL GRANT or REVOKE statements when granting or revoking permissions. In this case, the Active Directory user name should follow the DB2 user naming conventions.

DB2 and Delinea Plug-In Compatibility

Starting with DB2 release 10.5.4, DB2 does not allow security plug-ins to fork a process to authenticate DB2 users. To support this behavior, the Delinea plugins starting with Delinea for DB2 5.2.3 use the CentrifyDC service to authenticate Active Directory and local users.

If your environment contains DB2 10.5.4 or later:

- Only Delinea for DB2 5.2.3 or later plug-ins are supported, *and*
- Before you install Delinea for DB2, you must install the Server Suite 2015.1 or later agent (that is, agent version 5.2.3+) on each DB2 server, *and*
- You must ensure that the agent (that is, the centrifydc service) is running on each DB2 server.

Username-Password Plug-In

The Delinea username/password plug-in, centrifydc_db2userpass, supports authentication from both Active Directory and non-Active Directory users. A non-Active Directory user may be one of the following:

- a UNIX user from local stores such as /etc/passwd and Name Service Switch (NSS)
- any user who has been authenticated using Pluggable Authentication Modules (PAM)
- any user who has been authenticated using the AIX Loadable Authentication Module (LAM)

The Delinea username/password plug-in, like the IBM default username/password plug-in, gives you the option to allow users who are already logged in to a DB2 server machine to connect to a database instance without entering a user name or password. However, the default is to require a logged in user to re-enter the user name and password to access the database instance.

GSSAPI Plug-In

The GSSAPI plug-in, centrifydc_db2gsskrb5, supports single sign on to a DB2 instance using the user's Active Directory account. This plug-in assumes that the user requesting access to the database is already logged in to the client computer and has been authenticated through the Kerberos mechanism.

The GSSAPI plug-in allows users to run the connect and attach commands without specifying a user name and password even if the user is connecting from a remote DB2 client. It requires the user to have a valid Kerberos ticket. Generally, users obtain a Kerberos ticket automatically when they log in as an Active Directory user. However, in the following situations the user does not obtain a ticket automatically:

- The user logs in to the DB2 server as a local, non-Active Directory user.
- The user enters the UNIX command su - *user* as root to get a shell owned by another Active Directory user or local user.
- The user logs in as a user who has both an Active Directory account and a local user account. However, the Active Directory account is not in the same zone as the machine you logged in to.

In each of these cases, the user needs to obtain Kerberos tickets before single sign-on support is provided.

To obtain tickets for an Active Directory user, type kinit *user*. The user is prompted for a password. To avoid being prompted, you can create a keytab file in advance using the adkeytab command, set the environment variable KRB5_ KTNAME to the full path of your keytab file, and then run kinit -k *user@DOMAIN* to obtain the tickets.

> **Note**: If a user name is explicitly provided when only the GSSAPI plug-in is installed (for example, by entering the DB2 command connect to testdb user username using password), the plug-in first authenticates the given user to the Kerberos Key Distribution Center (KDC), and then obtains a ticket-granting ticket (TGT) upon success. The plug-in next uses the TGT to get a service ticket for the DB2 server.

Group Plug-In

You install the Group plug-in, centrifydc_db2group to retrieve the list of groups to which a user belongs for authorization. The group plug-in is called automatically after user authentication by DB2. The group info retrieved is used by DB2 to check a user's access rights and determine whether the user has privilege to do specific tasks; for example, connect, query, perform database management, and so on.

The Group plug-in queries Active Directory first for the groups to which the user belongs, and then it looks in the local groups on the host. The two lists are then merged, with duplicates removed and returned to DB2.

Make Connections to the DB2 Administration Server

The DB2 Administration Server (DAS) allows administrators to manage DB2 instances remotely. Using utilities such as DB2 Control Center (db2cc) to perform operations such as creating, removing, starting, or stopping a database instance remotely require a DAS connection. Tasks that can be performed on a running instance (such as creating or dropping a table in the instance) do not require a DAS connection.

The DAS uses a separate authentication scheme from the instance authentication. The DAS does not call into the DB2 security plug-ins or PAM when authenticating users. If you want to log in as an Active Directory user and use utilities such as DB2 Control Center to remotely administer an instance, you have the following options, irrespective of the plug-ins that you select:

- Run the utility that connects to the DAS (such as db2cc) on the DB2 server machine as the user who can perform the desired administrative tasks.

  Make sure that this user is in the same zone as the DB2 server machine.

- Install and configure either the Microsoft or Delinea password synchronization service. For more details about the Delinea password synchronization service, refer to the *Administrator's Guide for Windows*.

- Create a local user on the DB2 server machine and enter that user's user name and password when DB2 Control Center (or other utility connecting to the DAS) requests a user name and password.

Next Step:

Install and Configure the Server

This section describes how to install and configure the Authentication Service for IBM DB2 package on a DB2 server.

To automate Authentication Service for IBM DB2 plug-in installation and configuration, use the setupdb2.sh script provided in the Authentication Service for IBM DB2 package.

To manually install, set up, configure, and verify the Identity Broker Service for IBM DB2 plug-in without using the setupdb2.sh script, see Install Manually.

Use the uninstallation script, /usr/share/centrifydc/bin/ uninstalldb2.sh, included in the Authentication Service for IBM DB2 package, to remove the Authentication Service for IBM DB2:

- When there is partially installed Authentication Service for IBM DB2 release after a failed installation attempt.
- Before upgrading an existing Authentication Service for IBM DB2 to a new release.
- For details about using this script, see Execute the uninstalldb2 Script.

The following sections describe how to install and configure the Authentication Service for IBM DB2 package on each supported platform using the setupdb2.sh script:

- Software Requirements
- Unzip and Restore the Authentication Service for DB2 package
- Install Authentication Service for DB2 Using the Platform Install Program
- Install and Configure PlugIns Using the setupdb2.sh Script
- Install Manually
- Upgrade from an Earlier Release
- If an Installation Attempt Fails

Software Requirements

You must have the Delinea agent installed on each DB2 server, and the DB2 servers must be joined to an Active Directory domain.

If you use the GSSAPI plug-in, the plug-in must be installed on the DB2 server and each DB2 client. In addition, both the DB2 client and the DB2 server computers must be joined to the same Active Directory domain.

If you use the username/password plug-in, you must install the PAM library. You can install the PAM library after you install the Delinea for DB2 package.

See *DB2 and Delinea Plug-in Compatibility* under Authentication and Authorization in IBM DB2

See the release notes for the Delinea software, DB2 versions and versions of Red Hat, SuSE, Solaris, and AIX operating systems supported in this release. In general, the Delinea for DB2 package supports the same versions of Solaris, Red Hat, SuSE and AIX operating systems supported in DB2 version 9.5, 9.7, 10.1, and 10.5 with the following exception:

For Red Hat and SuSE Linux, only x86 and x86-64 bit (AMD style) architectures are supported.

Unzip and Restore the Authentication Service for DB2 Package

> **Note**: If Authentication Service for IBM DB2 is already installed, uninstall it now as described in Execute the uninstalldb2 Script.

To begin the installation, unzip and restore the Authentication Service for IBM DB2 package on each DB2 server.

Depending on the platform, download the plug-ins from the customer support portal.

The following sections describe how to unzip and restore the package on each supported platform. In each example:

- *release* is the release number of the Authentication Service for IBM DB2 software (for example, 4.5.0)
- *os_release* is the release number of the operating system (for example, 10.0)
- *architecture* is the processor architecture that is supported (for example, i386)

Unzip and Restore AIX Files

Execute the following commands to unzip and restore the Authentication Service for IBM DB2 package files on an AIX computer:

```
gunzip centrify-db2-release-aixos_release-ppc.tgz
```

```
tar -xvf centrify-db2-release-aixos_release-ppc-bff.tar
```

```
gunzip centrify-db2-release-aixos_release-ppc-bff.gz
```

After you execute these commands, the file centrify-db2-release-aixos_release-ppc-bff is ready to be installed using the native AIX installer. Go to Install Authentication Service for DB2 Using the Platform Install Program and continue from there.

Unzip and Restore Linux Files

Execute the following commands to unzip and restore the Authentication Service for IBM DB2 package files on a Linux computer. The examples shown here assume that you are installing on Red Hat Linux.

```
gunzip centrify-db2-release-rhelos_release-architecture.tgz
```

```
tar -xvf centrify-db2-release-rhelos_release-architecture.tar
```

After you execute these commands, the file centrify-db2-release-rhelos_release-architecture.rpm is ready to be installed using the native Linux installer. Go to Install Authentication Service for DB2 Using the Platform Install Program and continue from there.

Unzip and Restore Solaris files

Execute the following commands to unzip and restore the Authentication Service for IBM DB2 package files on a Solaris computer:

```
gunzip centrify-db2-release-solos_release-ppc-bff.tgz
```

```
tar -xvf centrify-db2-release-solos_release-ppc-bff.tar
```

After you execute these commands, the file centrify-db2-release-solos_release-ppc-bff is ready to be installed using the native Solaris installer. Go to Install Authentication Service for DB2 Using the Platform Install Program and continue from there.

Install Authentication Service for DB2 Using the Platform Install Program

After you have unzipped and restored the Authentication Service for IBM DB2 package files, install the package using the platform's native installation program. The following sections describe the installation procedure on each supported platform. In each example:

- *release* is the release number of the Authentication Service for IBM DB2 software
- *os_release* is the release number of the operating system.

Install the AIX Files

Execute the following command to install the Authentication Service for IBM DB2 package using the native AIX installation program:

```
installp -d centrify-db2-release-aixos_release-ppc-bff CentrifyDC.db2
```

After you execute this command, you are ready to install and configure the Authentication Service for IBM DB2 plug-ins. You can install and configure the plug-ins using the setupdb2.sh script, or manually without using the setupdb2.sh script. See Install and Configure PlugIns Using the setupdb2 Script or Install Manually for details about these procedures.

Install the Linux Files

Execute the following command to install the Authentication Service for IBM DB2 package using the native Linux installation program. The examples shown here assume that you are installing on Red Hat Linux.

If you are installing the Authentication Service for IBM DB2 package for the first time:

```
rpm -ivh centrify-db2-release-rhelos_release-architecture.rpm
```

After you execute this command, you are ready to install and configure the Authentication Service for IBM DB2 plug-ins. You can install and configure the plug-ins using the setupdb2.sh script, or manually without using the setupdb2.sh script. See the next section or Install Manually for details about these procedures.

Install the Solaris Files

Execute the following command to install the Authentication Service for IBM DB2 package using the native Solaris installation program.

```
pkgadd -a admin -n -d centrify-db2-release-solos_release.rpm
```

After you execute this command, you are ready to install and configure the Authentication Service for IBM DB2 plug-ins. You can install and configure the plug-ins using the setupdb2.sh script, or manually without using the setupdb2.sh script. See Install and Configure PlugIns Using the setupdb2 Script or Install Manually for details about these procedures.

Install and Configure Plug-Ins Using the setupdb2 Script

The /usr/share/centrifydc/bin/setupdb2.sh script is an interactive script. Provide the following information at the script prompts:

- The DB2 authentication you want to use (both user name/ password and single sign on, single sign on only, or username/ password only)
- What data sent to DB2 you want to encrypt
- The Active Directory administrator password

For GSSAPI-related plug-in installation using the setupdb2.sh script, additionally provide the following information at the prompts:

- An account name, password, and container for an Active Directory user with administrator privileges on the domain controller.

The scripts then installs, configures, and verifies the plug-in(s) according to your entries.

The following table lists the setupdb2.sh command line options:

inst	Yes	A string value	The name of a DB2 database instance.
verbose	No	0 or 1 The default is 1	If the value is 0, only the basic questions are asked. All 3 Authentication Service for IBM DB2 plug-ins are installed. If the value is 1, the script prompts for different installation and setup options.
debug	No	0 or 1 The default is 0	If the value is 0, installation and setup are performed. If the value is 1, the script simulates the steps without actually performing them. Each command is displayed with a "#" prefix. Use this option to preview what commands are executed in an actual invocation.

The format for all command options is option=value. Separate each option with a space.

Run the setupdb2.sh Script

Perform the steps described in this section to run the setupdb2.sh script now.

In the example used here, db2inst1 is the name of a DB2 database instance, you want to run the script in verbose mode, and you do not want to run the script in debug mode.

To run the setupdb2.sh script:

1. Change to the /usr/share/centrifydc/bin directory:

   ```
   cd /usr/share/centrifydc/bin
   ```

2. Run the setupdb2.sh script. The instance name that you specify with the setupdb2.sh command cannot exceed 8 bytes. In this example, the database instance is named db2inst1, verbose mode is invoked so that all prompts for different installation and setup options are displayed, and debug mode is not invoked.

   ```
   ./setupdb2.sh inst=db2inst1 verbose=1
   ```

   In this example, the database instance is named db2inst1, verbose mode is invoked so that all prompts for different installation and setup options are displayed, and debug mode is not invoked.

3. Type y or n at the prompt, Is db2inst1 a DB2 server install?

   In this example, db2inst1 is a server installation, so select the default (y for yes).

This is confirming that the running component is a DB2 server. Entering yes directs the script to also install the DB2 client component. A message indicates if the script determined the instance is 32 or 64 bit.

```
db2inst1 is a 64 bit instance. DB2 server and client setup will be done.
```

4. Enter a number at the prompt, Which DB2 auth method do you want to use?

   Select an authentication method. From the listed choices, enter the corresponding number.

   ```
   [1] Username/Password and Single sign-on
   [2] Single Sign-on only
   [3] Username/Password only
   [4] Skip this step
   Select a number from the menu [1]:
   ```

   See Username-Password Plug-In and GSSAPI Plug-In for details about these choices. In this example, select username/password only.

5. Enter a number at the prompt, Which data sent to DB2 should be encrypted?

   Select if or which data sent to DB2 should be encrypted. This step is optional.

   ```
   [1] Nothing
   [2] The username and their password
   [3] All data going to the server
   [4] Encrypt and compress all data going to the server
   [5] Skip this step
   ```

   In this example, select [1] Nothing. Selecting [2], [3], or [4] changes the SRVCON_AUTH to Server_ Encrypt. Selecting [5] Skip this step exits the plug-in setup program.

6. Type y or n at the prompt, Use the CentrifyDC group plugin?

   Specify whether to use the CentrifyDC group plug-in. See Group Plug-In for details about this choice.

   Install the Group plug-in centrifydc_db2group, to retrieve the list of groups to which a user belongs for authorization. The group plug-in is called automatically after user authentication by DB2.

   The group information retrieved is used by DB2 to check a user's access rights and determine whether the user has privilege to do specific tasks. For example: connect, query, db management, and so forth.

   The Group plug-in queries Active Directory first for the groups to which the user belongs and then it looks in the local groups on the host. The two lists are then merged with duplicates removed and returned to DB2.

   In this example, select yes.

7. Enter a number at the prompt, Do you want to configure the instance user db2inst1 as a service account?

   Specify whether to configure the instance user as a service account.

   You must do this step if you want to use the GSS-Plugin. If you already did this step for this instance, select the option to indicate the keytab file name.

   ```
   [1] Use`adkeytab` to create a service account in Active Directory and `keytab`file`.
   NOTE: You need to specify a user name with administrator privileges on the domain to use adkeytab.
   [2] Provide the name of an already existing`keytab`file.
   [3] Skip this step
   ```

   Generally, if you are starting from nothing, enter 1, otherwise enter 2.

   If you are setting up the GSSAPI plug-in (that is, if you selected a single sign-on option in Step 5) and you have not yet configured the instance user as a service account, you must select option 1, Use adkeytab to create a service account in Active Directory and keytab file. You will be prompted later for the Active Directory Administrator password.

If you have already configured the instance user as a service account, the necessary keytab file already exists. If this is the case, select option 2, "Provide the name of an already existing keytab file," and provide the full path and file name of the keytab file.

If you are not setting up the GSSAPI plug-in, you can optionally skip this step.

In this example, even though the GSSAPI plug-in is not being set up (that is, a single sign-on option was not selected in Step 5), you can still choose to configure the instance user as a service account. To do so, select option 1.

8. Enter a filename or press return to accept the default, at the prompt, What is the file name thatadkeytabshould use when creating thekeytabfile?

   Choose the default or specify any location.

   Full path please.

   Note: the file needs to be accessible to the db2inst1 user.

   [/home/db2inst1/db2inst1.keytab]

9. Enter at the prompt, Enter the password for db2inst1.

   Provide the password for the database instance that you specified in Step 2.

   Create a new password for db2inst1 or enter an existing password (if configured earlier).

10. Enter at the prompt, Enter a user name that has administrator privileges for the domain.

    Specify a user name (for example, flast@company.com). The username must be a SamAccount, and must have administrator privileges for the domain (that is, Active Directory Administrator privileges).

11. Enter at the prompt, Enter the container where to store the db2inst1 user.

    Specify the container object in which to create the service account.

    [CN=Users]:

    The default OU is CN=Users

    PAM setup not required for AIX. Skipping...

    If a service account name other than the DB2 instance name is chosen to adopt and build the Kerberos keytab file, this service account needs to meet the following two requirements:

    - The account name must be 8 characters long at most. This is required by the DB2 server.
    - This account must have the same permission granted as the instance owner in DB2 server

    The setupdb2.sh script can use only the container objects in the domain to which the computer is currently joined. You cannot specify another domain for the container object when you use the setupdb2.sh script to install and configure plug-ins. If you want to specify a different domain, you must install the plug-ins manually without using the setupdb2.sh script. See Step 2 in Set up for the GSSAPI plug-in for details about specifying a different domain.

    Type the name of the container object in relative DN format (that is, do not specify the domain portion of the DN). For example, if you wanted to create the service account in the users container in the currently joined domain, you would type the following:

    CN=users

12. Enter at the prompt, What group should be used as the group owner of this file?

    Specify the group name or select the default.

    All DB2 instances that you want to use the username/password plugin must be in this group.[db2iadm1]:

    You are prompted for more information depending on which plug-ins you are setting up:

    - The group that owns the /usr/share/centrifydc/bin/ db2userpass_checkpwd file. You are prompted for this information if you are setting up the username/password plug-in.
    - The password for the user with Active Directory Administrator privileges that you specified in Step 11. You are prompted for this information if you are setting up the GSSAPI plug-in.

    Example return output from this step.

********** `adkeytab` setup (required for GSS-plugin) **********

Using /home/db2inst1/db2inst1.keytab for the `keytab` file for instance: db2inst1

NOTE:`adkeytab`will prompt you for the password of the Active Directory admin user: rsriniva.

#`adkeytab`-n -c CN=Users -u rsriniva -K /home/db2inst1/ db2inst1.keytab -P db2inst1/vaix61-2.corp.contoso.com db2inst1 rsriniva@CORP.CONTOSO.COM's password:

Success: New Account: db2inst1

NOTE:`adkeytab`will prompt you for the password of the Active Directory admin user: rsriniva again.

#`adkeytab`-C db2inst1 -u rsriniva -w XXX-PASS-NOT-DISPLAYED- XXX -K /home/db2inst1/db2inst1.keytab rsriniva@CORP.CONTOSO.COM's password:

Success: Change Password: db2inst1

chmod 600 /home/db2inst1/db2inst1.keytab

chown db2inst1 /home/db2inst1/db2inst1.keytab # db2set DB2ENVLIST=KRB5_KTNAME

adkeytab setup successfully!

********** username/password plugin setup **********

chmod 750 /usr/share/centrifydc/bin/db2userpass_checkpwd

chown root:db2iadm1 /usr/share/centrifydc/bin/ db2userpass_checkpwd

chmod u+s /usr/share/centrifydc/bin/db2userpass_checkpwd username/password setup successfully

******* Installing the plugins into instance: db2inst1 *******

Installing client side auth plugin

rm -f sqllib/security32/plugin/client/ centrifydc_db2gsskrb5.so

cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so sqllib/security32/plugin/client/centrifydc_db2gsskrb5.so

Installing group plugin

rm -f sqllib/security32/plugin/group/centrifydc_db2group.so

cp /usr/share/centrifydc/lib/libcentrifydc_db2group.so sqllib/security32/plugin/group/centrifydc_db2group.so

Installing server side auth plugin

rm -f sqllib/security64/plugin/server/ centrifydc_db2gsskrb5.so

rm -f sqllib/security64/plugin/server/ centrifydc_db2userpass.so

cp /usr/share/centrifydc/lib64/libcentrifydc_db2gsskrb5.so sqllib/security64/plugin/server/centrifydc_db2gsskrb5.so

cp /usr/share/centrifydc/lib64/ libcentrifydc_db2userpass95.so sqllib/security64/plugin/ server/centrifydc_db2userpass.so

Installing client side auth plugin

rm -f sqllib/security64/plugin/client/ centrifydc_db2gsskrb5.so

cp /usr/share/centrifydc/lib64/libcentrifydc_db2gsskrb5.so sqllib/security64/plugin/client/centrifydc_db2gsskrb5.so

Installing group plugin

rm -f sqllib/security64/plugin/group/centrifydc_db2group.so

cp /usr/share/centrifydc/lib64/libcentrifydc_db2group.so sqllib/security64/plugin/group/centrifydc_db2group.so

******* Updating settings for DB2 instance: db2inst1 ******

Old configuration (You may want to copy these settings down in case you need to revert to the old settings):

Group Plugin (GROUP_PLUGIN) =

GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) = Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) = Server Userid-Password Plugin (SRVCON_PW_PLUGIN) = Server Connection Authentication (SRVCON_AUTH) =

NOT_SPECIFIED

Database manager authentication (AUTHENTICATION) = SERVER The DB2 configuration will be updated to:

LOCAL_GSSPLUGIN = centrifydc_db2gsskrb5 SRVCON_GSSPLUGIN_LIST = centrifydc_db2gsskrb5 SRVCON_PW_PLUGIN = centrifydc_db2userpass SRVCON_AUTH = GSS_SERVER_ENCRYPT AUTHENTICATION = SERVER

GROUP_PLUGIN = centrifydc_db2group

13. Review the script displayed content.

From this point the script stops the DB2 instance: db2inst1, updates the configuration, and then restarts the instance.

System information displays as files are configured. When the setupdb2.sh script finishes the configuration, a completion message displays.

Examples output when the instance is stopped.

```
Stopping instance: db2inst1
# db2stop
SQL1064N DB2STOP processing was successful.
# db2 update dbm config using LOCAL_GSSPLUGIN centrifydc_db2gsskrb5
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
# db2 update dbm config using SRVCON_GSSPLUGIN_LIST centrifydc_db2gsskrb5
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
# db2 update dbm config using SRVCON_PW_PLUGIN centrifydc_db2userpass
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
# db2 update dbm config using SRVCON_AUTH GSS_SERVER_ENCRYPT DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
# db2 update dbm config using AUTHENTICATION SERVER DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
# db2 update dbm config using GROUP_PLUGIN centrifydc_db2group DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
New configuration:
Group Plugin (GROUP_PLUGIN) = centrifydc_db2group
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) = centrifydc_db2gsskrb5
Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) = centrifydc_db2gsskrb5
Server Userid-Password Plugin (SRVCON_PW_PLUGIN) = centrifydc_db2userpass
Server Connection Authentication (SRVCON_AUTH) = GSS_SERVER_ENCRYPT
Database manager authentication (AUTHENTICATION) = SERVER
Starting Instance # db2start
SQL1063N DB2START processing was successful.
The plugins for DB2 instance: db2inst1 were setup successfully!
```

14. Verify if the setup completed properly or not by running the command as the DB2 instance user:

    ```
    db2 get dbm config |egrep -i "auth|gss|group|srvcon"
    ```

    Example of return output from the command for a scenario where all three DirectControl for DB2 security plug-ins have been configured as shown below. The lines of interest are in **bold**.

    ```
    SYSADM group name (SYSADM_GROUP) = DB2GRP1
    SYSCTRL group name (SYSCTRL_GROUP) = SYSMAINT group name (SYSMAINT_ GROUP) =
    SYSMON group name (SYSMON_GROUP) =
    __Group Plugin (GROUP_PLUGIN) = centrifydc_db2group__
    __GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) = centrifydc_ db2gsskrb5__
    __Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) = centrifydc_ db2gsskrb5__
    __Server Userid-Password Plugin (SRVCON_PW_PLUGIN) = centrifydc_ db2userpass__
    __Server Connection Authentication (SRVCON_AUTH) = SERVER_ENCRYPT__
    __Database manager authentication (AUTHENTICATION) = SERVER Cataloging allowed without authority (CATALOG_NOAUTH) = NO Trusted client authentication (TRUST_CLNTAUTH) = CLIENT__
    Bypass federated authentication (FED_NOAUTH) = NO
    ```

This completes the automated installation on the DB2 server. If you selected single sign-on and username/password or single sign-on only, you need to install the GSSAPI client on every client computer. Go to Set up the GSSAPI DB2 Client for information about that procedure.

If you selected username/password only, you are done with the installation. Go to Test the Installation to finish.

Install Manually

Perform the following steps if you want to install Authentication Service for IBM DB2 manually without using the setupdb2.sh script. If you already installed Authentication Service for IBM DB2, skip this section and go to Set up the GSSAPI DB2 Client.

- Perform the procedures described in Unzip and Restore the Authentication Service for DB2 package and Install Authentication Service for DB2 Using the Platform Install Program
- Copy the Authentication Service for IBM DB2 shared libraries to the appropriate DB2 locations. See Copy the plug-ins.
- If you plan to use username/password for authentication, configure the operating system to load the username/password plug libraries. See Set Up for the Username-Password Plug-in
- If you plan to use single sign-on, configure the operating system to use the GSSAPI plug-in and set up the key table. See Set up for the GSSAPI plug-in
- Configure DB2 to use the three plug-ins. See Configure the DB2 instance
- Confirm that the DB2 configuration is correct. See Verify the Setup

  Note: The Authentication Service for IBM DB2 Group plug-in requires no setup.

Copy the plug-ins

Use the following commands to copy the Authentication Service for IBM DB2 shared libraries from the installation directory to the proper DB2 directory for each instance—db2inst1 in the commands that follow.

The libcentrifydc_db2userpass.so that you use is version-dependent.

Copy Commands for 64-bit Instances:

```
cp /usr/share/centrifydc/lib64/libcentrifydc_db2userpass.so ~db2inst1/sqllib/security64/plugin/server/centrifydc_db2userpass.so
cp /usr/share/centrifydc/lib64/libcentrifydc_db2gsskrb5.so ~db2inst1/sqllib/security64/plugin/server/centrifydc_db2gsskrb5.so
cp /usr/share/centrifydc/lib64/libcentrifydc_db2gsskrb5.so ~db2inst1/sqllib/security64/plugin/client/centrifydc_db2gsskrb5.so
cp /usr/share/centrifydc/lib64/libcentrifydc_db2group.so ~db2inst1/sqllib/security64/plugin/group/centrifydc_db2group.so
cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so ~db2inst1/sqllib/security32/plugin/client/centrifydc_db2gsskrb5.so
```

Copy Commands for 32-bit Instances:

```
cp /usr/share/centrifydc/lib/libcentrifydc_db2userpass.so ~db2inst1/sqllib/security32/plugin/server/centrifydc_db2userpass.so
cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so ~db2inst1/sqllib/security32/plugin/server/centrifydc_db2gsskrb5.so
cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so ~db2inst1/sqllib/security32/plugin/client/centrifydc_db2gsskrb5.so
cp /usr/share/centrifydc/lib/libcentrifydc_db2group.so ~db2inst1/sqllib/security32/plugin/group/centrifydc_db2group.so
```

Setup for the Username-Password Plug-In

The username/password plug in library, centrifydc_db2userpass.so is now in place. Three more procedures are required to finish Authentication Service for IBM DB2 username/password plug-in installation and configuration:

- Configure the instance's Linux computer(s) to use the Authentication Service for IBM DB2 library for PAM authentication.

  Note: The Authentication Service for IBM DB2 username/password security plug-in uses PAM to authenticate users. This step is required only for DB2 servers running on Linux platforms. On AIX-based computers, the Authentication Service for IBM DB2 username/password plug-in uses the native LAM authentication framework which is already configured for authentication against Active Directory accounts.
- Set parameters in the /etc/centrifydc/centrifydc.conf file.
- Assign permissions for the program that checks the password for local users.

1. Configure Linux-based Computers:

Note: This operation requires root user privileges.

You need to tell the PAM service to use Authentication Service for IBM DB2 plug-in for authentication and account management. The name of the Authentication Service for IBM DB2 username/password plug-in is centrifydc_db2userpass.

Each PAM service has its own configuration file in the /etc/pam.d directory. To add the Authentication Service for IBM DB2 username/password plug-in on a Red Hat Linux computer, create the file

/etc/pam.d/centrifydc_db2userpass

with the following contents:

```
# Delinea PAM service for DB2 username/password support
# %PAM-1.0
auth required pam_stack.so service=system-auth
auth required pam_nologin.so
account required pam_stack.so service=system-auth
########################################
```

If you are configuring a SUSE Linux 10 computer, the contents of /etc/pam.d/centrifydc_db2userpass should be as follows:

```
auth include common-auth
account include common-account
```

If you are configuring a SUSE Linux 8 or 9 computer, the contents of /etc/pam.d/centrifydc_db2userpass should be as follows:

```
auth required pam_unix2.so
auth required pam_nologin.so
auth required pam_env.so
account required pam_unix2.so
account required pam_nologin.so
```

2. Set /etc/centrifydc/centrifydc.conf parameters

The following configuration options require you to edit the /etc/centrifydc/centrifydc.conf file on the DB2 server.

- If you want to allow users who are already logged in to the DB2 server to log in to the database instance without entering their user name and password, add the following line to /etc/centrifydc/centrifydc.conf:

  ```
  db2.userpass.allow.localnopasswd.db2_instance_name: true
  ```

  The default value is false, meaning that users already logged in to the server must enter their user name and password to access the database instance.

- If you have an environment in which the user name case used for database authentication differs from user name case stored in /etc/passwd, you need to add the following parameter to the /etc/centrifydc/centrifydc.conf file:

  ```
  db2.userpass.username.lower: true
  ```

  When this parameter is present and set to true, the DB2 username/password plug-in converts the user name to lowercase before attempting authentication. When this parameter is set to false, it leaves the case as-is.

- By default, the Delinea DB2 agent authenticates all Active Directory users even if the Active Directory user is not in the zone. To optionally constrain the authentication to zone enabled Active Directory users only, add the following parameter to the /etc/centrifydc/centrifydc.conf file:

  ```
  db2.user.zone_enabled.db2_instance_name: true
  ```

  After you add this parameter, restart the DB2 instance to pick up the new setting.

  Stop and start the agent after you modify centrifydc.conf to enable the conversion.

Set up for the GSSAPI PlugIn

This section describes how to configure the server to use the Authentication Service for IBM DB2 GSSAPI plug-in.

1. As root, use the adjoin command to join the UNIX DB2 server machine and each UNIX DB2 client using GSSAPI to the same Active Directory domain. See the *Administrator's Guide for Windows* for the adjoin command options. Be careful to join the appropriate Active Directory organizational unit and Delinea zone for your configuration.

   > **Note**: You must have the account name and password for an Active Directory user that has administrator privileges on the Active Directory domain controller to use adjoin. If you do not specify the account name in the adjoin command line you will be prompted to enter the administrator password.

2. As root, use the adkeytab command to create a Kerberos service account for the DB2 instance and generate a keytab file. (The adkeytab tool is included in the Delinea Server Suite package; see /usr/sbin.)

   The following example creates the account for the database instance db2inst1 in the Users container in the currently joined domain. The account resides on a DB2 server with host name (not fully-qualified) hostname, and generates a keytab file (db2inst1.keytab) in the $INSTHOME directory. Substitute your own instance, host, and keytab file names as appropriate.

   ```
   adkeytab -n -c CN=Users -u Administrator -K \
   $INSTHOME/db2inst1.keytab -P db2inst1/hostname db2inst1
   ```

   If you had wanted to create the account in a different domain than the currently joined domain, you would have used the adkeytab -d option.

   This example uses the domain controller's Administrator account to generate the keytab file and requires root to know the administrator password. If you do not know the administrator password, use the -u option to specify any user with administrator privileges on the Active Directory domain controller.

   The adkeytab command always sets the password of the domain account to a random value regardless of whether the account already exists. Use the following command to change the Active Directory password. This example uses db2inst1 for the DB2 instance name and password for the password string for the instance user's account in Active Directory. Substitute your own instance and password as appropriate.

   ```
   adkeytab -C db2inst1 -w password
   ```

   > **Note**: If there is a local user (for example, in /etc/passwd or /etc/shadow) with the same account name as the instance user, the adkeytab command does not change the local password.

   In both examples, you are prompted for the Active Directory Administrator password before the command is executed.

   After you have generated the keytab file with the adkeytab command, do not move or delete it. If you do, the agent will not renew the keytab.

   In addition, set the service account password in Active Directory to *never expire*.

3. Open the file /etc/centrifydc/user.ignore and add the instance user to the end of the file. (This file contains user names that are always treated as local—for example, root, mail, and daemon—when looking up user information.) This allows the instance user to log in as a local user to perform maintenance tasks.

4. Set appropriate permissions to protect the keytab file generated in Step 2.

   For the GSSAPI plug-in to work, the keytab file must be made readable by the DB2 instance owner. In addition, because the keytab file contains sensitive information such as the secret key associated with the DB2 instance service account, it should be properly protected. Execute the following commands as root to achieve this. The following example uses db2inst1 for the DB2 instance name and db2grp1 for the primary group of the instance user. Substitute your own instance and group names as appropriate.

   ```
   chmod 600 $INSTHOME/db2inst1.keytab
   chown db2inst1:db2grp1 $INSTHOME/db2inst1.keytab
   ```

5. Set up the DB2 environment variables to use the new keytab file. By default, DB2 uses the keytab file defined in the KRB5_KTNAME environment variable for authentication. The default is /etc/krb5.keytab. The following procedures describe how to set the variable for different UNIX shells. Perform the action as the DB2 instance owner, and replace db2inst1 with your actual instance name.

   For Bourne, Korn and bash shell users, add the following lines to $INSTHOME/sqllib/userprofile:

   ```
   KRB5_KTNAME=$INSTHOME/db2inst1.keytab
   export KRB5_KTNAME
   ```

   For C shell users, add the following line to $INSTHOME/sqllib/usercshrc:

```
setenv KRB5_KTNAME $INSTHOME/db2inst1.keytab
```

By default, DB2 filters out all user environment variables except for those prefixed with DB2 or db2. To pass the value stored in KRB5_KTNAME to the DB2 instance, the variable must be added to the DB2ENVLIST parameter. To do so, run the following command as the DB2 instance user: db2set DB2ENVLIST=KRB5_KTNAME

Note: Before executing db2set, you must either:

- Log out after updating the userprofile and usercshrc files to set the KRB5_KTNAME environment and log back in again; or
- Set the environment variable in your shell before issuing the command.

6. On some platforms, the DB2 server may not be able to start due to the Kerberos library conflict between the system and Delinea DirectControl.

   The centrifydc_db2gsskrb5 plugin has to be linked against the one from DirectControl.

   To work around this issue, Delinea recommends that you add the library search path and make it readable by DB2 server. Please revise the changes above and do the modification as below:

   Note: The environment variable name and value for library search path is platform specific, and the example below is for Linux x86_64.

   For Bourne, Korn and bash shell users, add the following lines to $INSTHOME/sqllib/userprofile:

   ```
   KRB5_KTNAME=$INSTHOME/db2inst1.keytab
   export KRB5_KTNAME
   LD_LIBRARY_PATH=/usr/share/centrifydc/lib64:/usr/share/centrifydc/kerberos/lib64:$LD_LIBRARY_PATH
   export LD_LIBRARY_PATH
   ```

   For C shell users, add the following line to $INSTHOME/sqllib/usercshrc:

   ```
   setenv KRB5_KTNAME $INSTHOME/db2inst1.keytab
   setenv LD_LIBRARY_PATH=/usr/share/centrifydc/lib64:/usr/share/centrifydc/kerberos/lib64:$LD_LIBRARY_PATH
   ```

Run the following command as the DB2 instance user:

```
db2set DB2ENVLIST="KRB5_KTNAME LD_LIBRARY_PATH"
```

Configure the DB2 Instance

Enter the following commands to modify each DB2 instance's configuration parameters to use the Authentication Service for IBM DB2 plug-ins for authentication and authorization.

All of the following commands should be executed as an instance user.

Case 1: Use the username/password plug-in only:

```
db2 update dbm cfg using SRVCON_PW_PLUGIN centrifydc_db2userpass
db2 update dbm cfg using SRVCON_AUTH NOT_SPECIFIED
db2 update dbm cfg using AUTHENTICATION SERVER
```

Note: If you select the SRVCON_AUTH option, the user name and password are transmitted in the clear. This library also includes the following options to encrypt different parts of the message:

- SERVER_ENCRYPT: The user name and password are encrypted in messages sent from DB2 client to DB2 server.
- DATA_ENCRYPT: User data as well as the authentication data (user name and password) are encrypted in messages sent from DB2 client to DB2 server.
- DATA_ENCRYPT_CMP: DATA_ENCRYPT with backwards compatibility to older versions of the DB2 client. (If you have an older version of the DB2 client that does not support the DATA_ENCRYPT option, only the authentication data is encrypted unless you select the DATA_ENCRYPT_CMP option.)

For example, to set the username/password plug-in to encrypt all data going to the server you would use the following command:

```
db2 update dbm cfg using SRVCON_AUTH DATA_ENCRYPT
```

Case 2: Use the GSSAPI plug-in only:

```
db2 update dbm cfg using SRVCON_PW_PLUGIN NULL
db2 update dbm cfg using SRVCON_GSSPLUGIN_LIST centrifydc_db2gsskrb5
db2 update dbm cfg using LOCAL_GSSPLUGIN centrifydc_db2gsskrb5
db2 update dbm cfg using SRVCON_AUTH GSSPLUGIN
db2 update dbm cfg using AUTHENTICATION SERVER
```

Case 3: Use the username/password plug-in and the GSSAPI plug-in together:

```
db2 update dbm cfg using SRVCON_PW_PLUGIN centrifydc_db2userpass
db2 update dbm cfg using SRVCON_GSSPLUGIN_LIST centrifydc_db2gsskrb5
db2 update dbm cfg using LOCAL_GSSPLUGIN centrifydc_db2gsskrb5
db2 update dbm cfg using SRVCON_AUTH GSS_SERVER_ENCRYPT
db2 update dbm cfg using AUTHENTICATION SERVER
```

For all cases: Run the following command as the DB2 instance user to configure the instance to use the Authentication Service for IBM DB2 group plug-in:

```
db2 update dbmcfg using GROUP_PLUGIN centrifydc_db2group
```

This completes the Authentication Service for IBM DB2 package manual installation and configuration. Next, verify that the configuration parameters are set properly.

Verify the Setup

Execute the following command as the DB2 instance user to verify the setup:

```
db2 get dbm config legrep -i "authlgsslgrouplsrvcon"
```

A sample output of this command for a scenario where all three Authentication Service for IBM DB2 security plug-ins have been configured is as follows. The lines of interest are in **bold**.

```
SYSADM group name (SYSADM_GROUP) = DB2GRP1
SYSCTRL group name (SYSCTRL_GROUP) =
SYSMAINT group name (SYSMAINT_GROUP) =
SYSMON group name (SYSMON_GROUP) =
**Group Plugin (GROUP_PLUGIN) = centrifydc_db2group**
__GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) = centrifydc_ db2gsskrb5__
__Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) = centrifydc_ db2gsskrb5__
__Server Userid-Password Plugin (SRVCON_PW_PLUGIN) = centrifydc_ db2userpass__
**Server Connection Authentication (SRVCON_AUTH) = GSS_SERVER_ENCRYPT**
**Database manager authentication (AUTHENTICATION) = SERVER**
Cataloging allowed without authority (CATALOG_NOAUTH) = NO
Trusted client authentication (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication (FED_NOAUTH) = NO
```

After installing the plug-ins, the database instance needs to be stopped and restarted. Enter the `db2stop` and `db2start` commands as the instance user.

Upgrade from an Earlier Release

If you are upgrading from an earlier release of Authentication Service for IBM DB2, you have to stop the DB2 instance before the upgrade by using the `db2stop` command. After stopping the DB2 instance, you can upgrade using the `setupdb2.sh` script, or manually by copying the new plug-ins into their corresponding DB2 directories.

Upgrade Using the setupdb2.sh Script

1. Ensure that you have stopped the DB2 instance.

2. Remove the Authentication Service for IBM DB2 software as described in Uninstall DB2 Plug-ins
3. Install the new release of the Authentication Service for IBM DB2 package as described in Install and Configure Server

Upgrade Manually

1. Ensure that you have stopped the DB2 instance.
2. Remove the Authentication Service for IBM DB2 software as described in Uninstall DB2 Plug-ins
3. Perform the procedures described in Install Manually.
4. Restart the DB2 instance after the files are in place using db2start.

   If you are currently using a Beta version of the software, refer to Delinea Knowledge Base article KB-0938 for information about how to perform the upgrade.

If an Installation Attempt Fails

If you attempt to install the Authentication Service for IBM DB2 package and the installation fails, before retrying the installation you must uninstall any files that were installed by performing the procedures described in Uninstall DB2 Plug-ins

Next Step:

Set Up the GSSAPI Client

The Authentication Service for IBM DB2 GSSAPI security plug-in has a client component that must be installed on each DB2 UNIX-based client computer accessing the DB2 server.

DB2 Client Installation on a UNIX Computer

Copy the Delinea for DB2 package to each client. Unzip, restore, and install the package as described in Install and Configure Server.

Just like the DB2 server, you can use either use the setupdb2.sh setup script or manually install and configure the software. The following sections describe these procedures.

Install on UNIX Using the

To install the Delinea for DB2 package using the setupdb2.sh script, perform the steps described in Install and Configure Plug-ins Using the setupdb2.sh Script

> **Note**: The setupdb2.sh script may wrongly identify a DB2 version 8 client as a DB2 server. If this happens, when the script prompts you to confirm the detection, answer **no**. The script will then install the GSSAPI plug-in for DB2 client.

Install on UNIX Manually

Perform the following steps to install the Delinea for DB2 package manually.

To install the Authentication Service for IBM DB2 manually:

1. Copy the shared libraries. Run the following commands as the instance user to copy the shared libraries to the target directories where db2inst1 is the instance name:
   - For a 64-bit DB2 instance:

```
cp /usr/share/centrifydc/lib64/libcentrifydc_db2gsskrb5.so
~db2inst1/sqllib/security64/plugin/client/centrifydc_db2gsskrb5.so
cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so
~db2inst1/sqllib/security32/plugin/client/centrifydc_db2gsskrb5.so
```

   - For 32-bit instances, run:

```
cp /usr/share/centrifydc/lib/libcentrifydc_db2gsskrb5.so
~db2inst1/sqllib/security32/plugin/client/centrifydc_db2gsskrb5.so
```

2. Set up the DB2 configuration variables. As the DB2 instance user, run the following commands to tell DB2 to use server authentication schemes:

```
db2 update dbm cfg using LOCAL_GSSPLUGIN centrifydc_db2gsskrb5
db2 update dbm cfg using AUTHENTICATION SERVER
```

3. On some platforms, the DB2 client may not be able to run due to the Kerberos library conflict between the system and Delinea DirectControl.

   The centrifydc_db2gsskrb5 plugin has to be linked against the one from DirectControl.

   To work around the issue, Delinea recommends that you add the library search path. Please do the modification as below:

   > **Note**: The environment variable name and value for library search path is platform-specific, and the example below is for Linux x86_64.

   For Bourne, Korn and bash shell users, add the following lines to $INSTHOME/sqllib/userprofile:

```
LD_LIBRARY_PATH=/usr/share/centrifydc/lib64:/usr/share/centrifydc/kerberos/lib64:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

   For C shell users, add the following line to $INSTHOME/sqllib/usercshrc:

```
setenv LD_LIBRARY_PATH=/usr/share/centrifydc/lib64:/usr/share/centrifydc/kerberos/lib64:$LD_LIBRARY_PATH
```

This section describes how to test the Authentication Service for IBM DB2 security plug-ins after installation. The test can be performed on the DB2 server or from a computer with DB2 client software installed.

The procedure described below shows how an Active Directory user accesses a DB2 instance. The user joe is the Active Directory user in the same zone as the DB2 computer from which the test is executed. The DB2 database instance name is db2inst1, and sample is the database.

1. Grant the user joe access to select from a table. Log in as the instance user and run the following commands to grant user joe the right to select from the PROJECT table in the sample database:

   ```
   db2 connect to sample
   db2 GRANT SELECT on PROJECT to USER joe
   db2 terminate
   ```

2. After joe logs in, he should set up the environment variables before connecting to the database. From the shell prompt:
   - Set the INSTHOME environment variable to the home directory of the instance user. For example, if you are using Bourne shell or equivalent, type:

     ```
     eval export INSTHOME=~db2inst1
     ```

   - Set up the database environment using the following commands.

     Bourne shell or equivalent:

     ```
     . $INSTHOME/sqllib/db2profile
     ```

     C-shell or equivalent:

     ```
     source $INSTHOME/sqllib/db2cshrc
     ```

   These commands can also be added to joe's login script such as

   .cshrc or .profile

3. Connect to the sample database as joe using the following commands:
   - To test single sign-on, type:

     ```
     db2 connect to sample
     ```

   - To test connecting using user name and password, type:

     ```
     db2 connect to sample user joe
     ```

     You should see output similar to the following:

     ```
     Database Connection Information
     Database server = DB2/Linux 9.0
     SQL authorization ID = JOE
     Local database alias = SAMPLE
     ```

4. Verify that the database is functioning by querying the PROJECT table:

   ```
   db2 select '*' from db2inst1.project
   ```

Perform the uninstallation procedures described in the following sections on each instance from which you want to remove the Authentication Service for IBM DB2 plug-in software. You must perform these procedures before you upgrade to a new Authentication Service for IBM DB2 release, or to remove a partially installed Authentication Service for IBM DB2 release after a failed installation attempt.

The uninstallation procedures are as follows. Unless otherwise noted, each procedure is required.

- **Execute the uninstalldb2.sh Script** on DB2 clients and servers to revert DB2 to the settings that existed before the Authentication Service for IBM DB2 package was installed.
- **Manually Reset DB2 Configuration Variables**. This procedure is optional. In most situations, the uninstalldb2.sh script automatically resets DB2 configuration variables to their default values, or to the values that they had before the Authentication Service for IBM DB2 package was installed.

Execute the uninstalldb2 Script

The uninstallation script /usr/share/centrifydc/bin/uninstalldb2.sh will undo the Authentication Service for IBM DB2 installation and revert DB2 to its previous settings. The uninstalldb2.sh script can be run on a DB2 client and a DB2 server. The following table lists the uninstalldb2.sh options.

inst	Yes	A string value	The name of a DB2 database instance.
verbose	No	0 or 1 The default is 1	If the value is 0, only the basic questions are asked. All 3 Authentication Service for IBM DB2 plug-ins are uninstalled. If the value is 1, the script prompts for different options, such as which plug-ins to remove.
debug	No	0 or 1 The default is 0	If the value is 0, uninstallation is performed. If the value is 1, the script displays the steps without actually performing them. Each command is displayed with a "#" prefix. Use this option to preview what commands are executed in an actual invocation.

Because the inst option is required, you must know the name of the instance from which you are removing the Authentication Service for IBM DB2 plug-in software. The following section describes how to determine the instance name.

Determine the Instance Name

You can determine the instance name in one of these ways:

- By reviewing the DB2 log in this location:

  $INSTHOME/sqllib/db2dump/

- By executing one of the following commands:

On AIX:

```
/opt/IBM/db2/V9.5/instance/db2ilist
/opt/IBM/db2/V9.5/instance/db2ilist
```

On Linux:

```
/opt/ibm/db2/V9.5/instance/db2ilist
/opt/ibm/db2/V9.7/instance/db2ilist
```

Run the uninstalldb2.sh Script

The format for script options is *option= value*. In the following example, db2inst1 is the name of a DB2 database instance, the verbose option is selected, and the debug mode is not invoked:

```
uninstalldb2.sh inst=db2inst1 verbose=1
```

Execute the `uninstalldb2.sh` script now using options that are appropriate for your DB2 server or client.

Manually Reset DB2 Configuration Variables

> **Note**: This procedure is optional, as configuration variables are typically reset by the `uninstalldb2.sh` script.

Perform the procedure described in this section to manually set the DB2 configuration variables back to the values they had before the plug-ins were installed. If you know the original plug-in values, reset them accordingly.

If you do not know the original values, use the following commands to reset the variables to their default values. Run these commands as the instance owner:

```
db2 update dbm cfg using SRVCON_AUTH NOT_SPECIFIED
db2 update dbm cfg using AUTHENTICATION SERVER
db2 update dbm cfg using GROUP_PLUGIN NULL
db2 update dbm cfg using LOCAL_GSSPLUGIN NULL
db2 update dbm cfg using SRVCON_GSSPLUGIN_LIST NULL
db2 update dbm cfg using SRVCON_PW_PLUGIN NULL
```

> **Note**: These commands work for both a DB2 client and a DB2 server.

References

For further information about setting up DB2, see the following documentation:

- Quick Beginnings for DB2 Servers
- DB2 DB2 UDB Security Part 1: Understand how user and group accounts interact with DB2 UDB

This section describes how to adopt a service account and the permission required depends on the option chosen.

In a Kerberized environment, there are times when a service account needs to obtain a Kerberos credential and infinitely renew that credential for a long running process.

Another scenario configuring a clustered environment where a virtual host account needs to provide services using an additional ServicePrincipalName (SPN).

One way to achieve goals such as, but not limited to, the above scenarios, is to use the Delinea command adkeytab to adopt a service account and build a keytab file.

Option 1: Reset the Service Account Password

Let the adkeytab command reset this service account's password while adopting the account. The current password of the service account is not required.

With this option, the account adopting the service account needs to have reset password and change password permission of the service account. For example:

```
adkeytab --adopt -u svcadmin -K /etc/svcacct.keytab svcacct
```

From the example, the account svcadmin is performing the adoption so it must have permission to reset password and change password for the adopted account svcacct. After the adoption, the password of this service account, svcacct, is reset to a randomly generated password.

Option 2: Provide the Existing Service Account Password

Provide this service account's current password with adkeytab command while adopting the account. The current password for this service account is required.

With this option, the account adopting the service account does not need any extra permission; the default read permission is enough. With this option, the -local and -w flags are required to adopt this account. For example:

```
adkeytab --adopt -u svcadmin --local -w password> -K /etc/ svcacct.keytab svcacct
```

where <password> is replaced by account svcacct's current password. After the adoption, the password of this service account is not changed or reset.

See the adkeytab main page for a complete list of options and description.

The *Delinea-enabled PuTTY User's Guide* describes how to install and configure the Delinea-enabled PuTTY program on Windows computers. PuTTY is open-source client software that enables you to open telnet, secure shell, rlogin and raw TCP sessions on remote computers. The PuTTY client available in Server Suite has been modified to support Kerberos-based authentication on remote computers that are managed by Server Suite software.

Intended Audience

This guide is intended for users who want to use the Delinea-enabled PuTTY client to open sessions on remote computers and have their identity authenticated using their Kerberos credentials. This guide assumes that you are familiar with Server Suite components and that you have sufficient privileges to perform administrative tasks on your managed computers.

PuTTY is free open-source software that enables you to connect to remote computers using network protocols such as telnet, ssh, rlogin or raw TCP. The version of PuTTY that is widely available, however, does not support Kerberos authentication. The version of PuTTY that is available in Server Suite has been modified to enable users to be authenticated using their Kerberos credentials before establishing a remote connection.

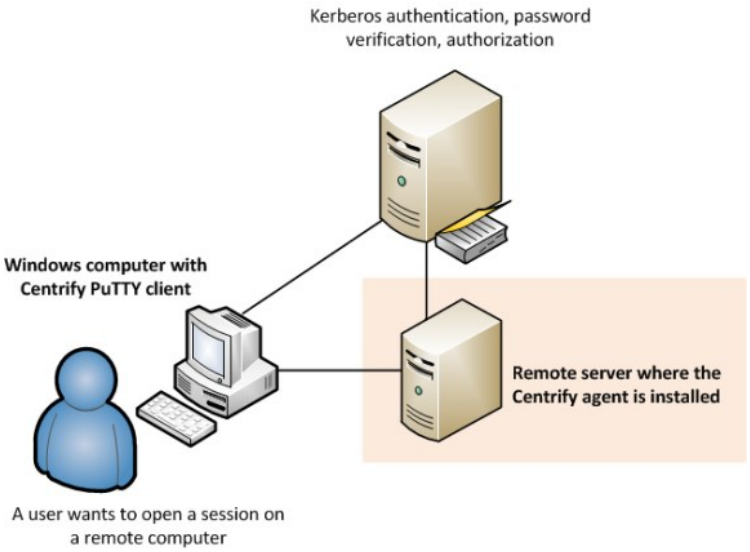
Accessing Remote Server Suite-Managed Computers

You can use the Centrify version of the PuTTY client with any supported protocol and to remotely access any Linux, UNIX, or Windows computer on your network, including computers that are not managed by the Server Suite Agent. However, the most common reason for using the Delinea PuTTY client is to open secure shell (ssh) sessions on remote Server Suite-managed computers. If you have the Server Suite Agent and Centrify OpenSSH installed on a remote computer, you can securely access that computer using your Active Directory credentials and take full advantage of centralized Kerberos authentication and consistent password policies across platforms.

If you use the Delinea PuTTY client to access Server Suite-managed computers through SSH, the Server Suite Agent can determine the UNIX login name to use from the user principal name (UPN) in Active Directory, making it possible for you to connect to any managed computers with a single Active Directory identity.

The Server Suite Agent is also responsible for setting up and managing the Kerberos environment on Server Suite-managed computers. You are not required to configure any DNSto-realm mapping because the agent already knows the relationship between the host computers and their service principal names (SPNs).

Because the Server Suite Agent automatically manages the Kerberos authentication and policy enforcement on Server Suite-managed computers, you can use the Delinea PuTTY client to connect to those computers using a secure and well-established authentication, authorization, and policy enforcement infrastructure.



If you use the Delinea PuTTY client with other protocols or to access remote computers that are not managed by the Server Suite Agent, the program operates in the same way as the standard PuTTY client. You can configure connections for other protocols and set other configuration options as you would for the open-source PuTTY client.

> **Note**: The Delinea PuTTY client is based on PuTTY version 0.64. This version of the Delinea PuTTY client is compatible with the Server Suite Agent, version 4.x and later, and with Centrify OpenSSH, version 4.x, and later.

Installing Delinea PuTTY

The Delinea PuTTY client software is only supported on Windows computers. Before installing, you should verify that you have a supported version of one of the Windows operating system product families. For example, you can use Windows 7 or Windows 8. Alternatively, you can install on computers in the Windows Server product family—such as Windows Server 2008 R2 or Windows Server 2012—if you want your computer to be configured with additional server roles.

For more detailed and most up-to-date information about supported operating system versions, see the Centrify website.

You can install the Delinea PuTTY client by selecting it when you install other Server Suite components or as a standalone executable using its own setup program. If you downloaded the Delinea PuTTY client as a separate software package from the Centrify website, the package includes the standalone setup program for installing the PuTTY client outside of Server Suite.

To install the Delinea PuTTY client from its standalone setup program

1. Double click on the putty-*version*.msi file to start the PuTTY client setup program.

   If another version of the software is installed on the local computer, you are prompted to remove it before you can proceed.

2. On the Welcome page, click **Next**.

3. Select a folder where the software should be installed by accepting the default location or clicking **Browse** to select a different location and specify who can use the PuTTY client on this computer. then click **Next**.

4. On the Confirm installation page, click **Next** to start the installation.

5. If you see a User Account Control warning, click Yes to continue.

6. Click **Finish** upon successful completion of the installation.

In addition to the PuTTY client (putty.exe), the following PuTTY-related programs are installed:

- pageant.exe is a secure shell (ssh) authentication agent for the PuTTY, PSCP, and Plink programs.
- plink.exe is a command-line interface to the PuTTY backend.
- pscp.exe is a command-line secure file copy (SCP) client.
- psftp.exe is a secure file transfer (SFTP) client.
- puttygen.exe is an RSA and DSA key generation utility.
- puttytel.exe is a Telnet-only client.

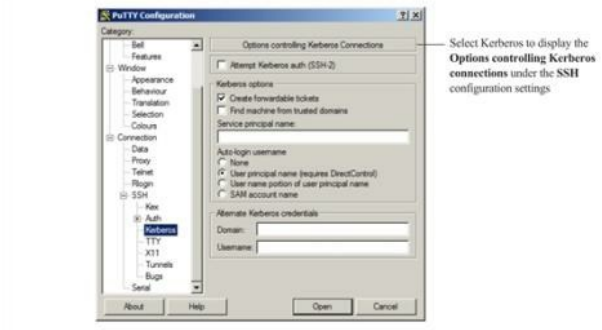
For more information about using these programs, see the official PuTTY documentation. For references to the official PuTTY documentation, see Getting More Information.

Configuring the Delinea PuTTY Client

The Centrify-enabled version of the open-source PuTTY client adds Kerberos authentication for accessing remote computers using secure shell (ssh) network connections. To enable you to configure Kerberos authentication for secure shell sessions, the Delinea PuTTY client adds its own SSH Kerberos configuration page to the standard Windows PuTTY client. All other functionality in the Delinea PuTTY client is the same as in the official PuTTY client, version 0.64.

Starting the Delinea PuTTY Client

After installation, you can start the Delinea PuTTY client from the Start menu or by opening the putty.exe executable in the file location you specified during installation. By default, the **Basic options for your PuTTY session** are displayed. These options are the same in the Delinea PuTTY client as they are in the open-source PuTTY client. For example:

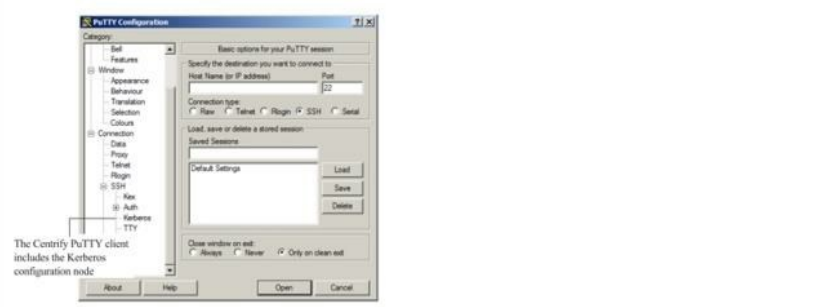


Configuring Kerberos Authentication for Secure Shell Connections

The Kerberos configuration options that have been added to the Centrify version of the PuTTY client are available under the Connection and SSH configuration settings.

To configure Kerberos settings:

1. Expand SSH under the Connection configuration settings. For example:



2. Select **Kerberos** to display the Options for controlling Kerberos connections. For example:

3. Set the appropriate options to configure Kerberos authentication for secure shell remote connections.
    - Select **Attempt Kerberos Auth (SSH-2)** if you want the Delinea PuTTY client to attempt to use Kerberos authentication before any other authentication method when opening a new secure shell session.

      If you do not select this option or select this option and Kerberos authentication fails, the authentication options you have defined in Connection > SSH > Auth are used. The number of times you can type the wrong password before Kerberos authentication fails and other authentication options are used can be configured by group policy settings. For more information about the group policies for configuring Delinea PuTTY, see Configuring Group Policies for Delinea PuTTY.

    - Select **Create forwardable tickets** if you want to allow the same Kerberos credentials used for authentication when connecting to other Kerberos-authenticated services.

      The option is selected by default to enable single sign-on, allowing you to be authenticated silently on other servers without providing a password. If you deselect this option, you are prompted to provide a password any time you connect to another Kerberos-authenticated service.

    - Select **Find machine from trusted domains** if you want the Delinea PuTTY client to look for computers in external trusted domains if it cannot locate a target computer in the local Active Directory forest or a trusted forest.

      If you select this option and the Delinea PuTTY client cannot locate a target computer, the program will attempt an LDAP connection to the domain controller in the trusted domains using your login credentials. The LDAP connection can only succeed if the domain controller is accessible and you have Read access in Active Directory. You can control the LDAP connection setting by using Delinea PuTTY group policies. For more information about the group policies for configuring Delinea PuTTY, see Configuring Group Policies for Delinea PuTTY.

        - Type a specific **Service principal name** if a target computer is in a different forest or if the Delinea PuTTY client cannot access the Kerberos Distribution Center (KDC) for the computer.

- You might have to specify the service principal name if a computer is located in an external trusted domain that is not accessible. For example, if firewall settings prevent the Delinea PuTTY client from making an LDAP connection to the domain controller in the trusted domains, you can explicitly identify the computer by its service principal name.

4. Select an **Auto-login username** option to specify how the Delinea PuTTY client determines the UNIX user account name to use for authentication when opening a secure shell connection.
   - Select **None** if you want to be prompted to specify the user name for Kerberos authentication or if you want to set a default auto-login user name as a Connection > Data configuration option.

     If you select this option, the Delinea PuTTY client does not automatically generate the UNIX user account name.
   - Select **User principal name (requires DirectControl)** if you want the Delinea PuTTY client to use your user principal name (UPN) as the UNIX account name.

     This option requires the Centrify Agent to be installed. With this option, the agent automatically maps the UPN in the Kerberos ticket to the UNIX profile for the Active Directory user name presented in the ticket.
   - Select **User name portion of user principal name** if you want the Delinea PuTTY client to use the user name portion of the UPN as the UNIX user name.

     If you select this option and the UPN is jdoe@xyz.com, the Delinea PuTTY client would use jdoe as the UNIX user name for authentication.
   - Select **SAM account name** if you want the Delinea PuTTY client to look up the sAMAccountName attribute in Active Directory and use it as the UNIX user name.

     If you select this option, the Delinea PuTTY client will initiate an LDAP connection to the currently logged-in domain controller. If the connection or lookup request fails, the Delinea PuTTY client will prompt you to enter the UNIX user name.
5. Type a **Domain** and **Username** if you do not want to use the Kerberos credentials for the account you used to log on to the Windows computer where you are running the Delinea PuTTY client.

   By default, your current Kerberos credentials for your Windows account are used for authentication on the remote computer. If you want to use a different user name and password, specify the domain and user name for the alternate Kerberos credentials you want to use. When the Delinea PuTTY client opens the secure shell session on the remote computer, it will prompt you to provide the password for your alternate credentials.

   The ability to use alternate Kerberos credentials can be configured by group policy settings. For more information about the group policies for configuring Delinea PuTTY, see Configuring Group Policies for Delinea PuTTY.

Saving and Managing Passwords for Remote Sessions

By default, the Kerberos credentials for the Active Directory account you used to log on to the Windows computer are used for authentication on remote computers. If the remote computer is found and authentication is successful, you are not prompted to provide a password.

If you open a secure shell session using alternate Kerberos credentials or the Delinea PuTTY client cannot locate the target computer using the Kerberos credentials you provided, it will prompt you to provide the new credentials.

If you are prompted for a password, you can select **Remember my password** to have your password stored in the Windows credential cache the password so that you are not prompted for again the next time you access the same remote computer. By saving your password or your user name and password in the Windows credential cache, you can have single sign-on (SSO) access to remote UNIX and Linux computers using your Active Directory user credentials.

If the Delinea PuTTY client cannot find the computer you specify using your own or the alternate Kerberos credentials you have specified, you can try other credentials or other configuration options, such as **Find machine from trusted domains**. If the new credentials or configuration options are successful, you can then select Remember my password to access that computer the next time you open a connection to it. After saving your information, you can use single sign-on to access computers in external or untrusted forests or in disjointed domains.

You can manage cached passwords by using the Credential Manager Control Panel or by opening a Command Prompt window and typing control keymgr.dll.

The number of times you can type the wrong password before Kerberos authentication fails and other authentication options are used can be configured by group policy settings. For more information about the group policies for configuring Delinea PuTTY, see Configuring Group Policies for Delinea PuTTY.

Configuring Group Policies for Delinea PuTTY

Centrify provides group policy administrative templates that allow you to centrally manage the configurable PuTTY settings for Kerberos authentication using secure shell connections. The group policy administrative templates are available in both admx and xml file formats.

- The admx template, centrify_putty_settings.admx, is installed by default in the C:\Windows\PolicyDefinitions directory.
- The xml file, centrify_putty_settings.xml, is installed by default in the same directory as the Delinea PuTTY program. For example, if you used the default location in the setup program, the file is located in C:\Program Files (x86)\Centrify\Centrify PuTTY.

To use group policies to configure Delinea PuTTY settings, an administrator must copy either the admx file or the xml file to the appropriate domain controller. If your organization centrally manages Delinea PuTTY settings through these group policies, you do not have to configure them manually for individual secure shell sessions.

By default, all group policies are set to **Not Configured**. Individual policies must be set to **Enabled** to activate a setting. Policies can also be set to **Disabled** to explicitly disable a setting. For details about how policies with Enabled or Disabled settings are inherited or overridden based on where they are applied, see the *Group Policy Guide* and Microsoft documentation for group policies.

Most group policy settings are equivalent to the configuration settings described in Configuring the Delinea PuTTY client. For more information about the opensource PuTTY client configuration settings, see the standard PuTTY documentation. For information about specific group policies, select the group policy, right-click to select **Properties**, then click the **Explain** tab.

Using Other Centrify-Enabled PuTTY Programs

In addition to the main PuTTy client (putty.exe), Centrify has modified the standard versions of the pscp.exe, psftp.exe, and plink.exe programs to support Kerberos authentication.

The modified pscp.exe program supports the following command formats:

```
pscp [options] [user@]host:source target
pscp [options] source [source...] [user@]host:target
pscp [options] -ls [user@]host:filespec
```

The modified psftp.exe program supports the following command formats:

```
psftp [options] [user@]host
```

The modified plink.exe program supports the following command formats:

```
plink [options] [user@]host [command]
```

Many of the PuTTY settings can be provided as options to the command line tools. You can also save command line settings into sessions and load them when executing commands using the –load option. If the settings in a saved session conflict with those specified when invoking the command, the specified options take precedence.

In addition to the standard PuTTY command line options, Delinea PuTTY provides the following options:

-k	Use Kerberos authentication and provide a UNIX user account name during login. This option is equivalent to selecting **Attempt Kerberos auth (SSH-2)** and **None** for the Auto-login username in the Delinea PuTTY Kerberos configuration page.
-K	Use Kerberos authentication and do auto login. This option is equivalent to selecting both **Attempt Kerberos auth (SSH-2**) and the **User principal name (requires DirectControl)** for the Auto-login username in the Delinea PuTTY Kerberos configuration page.
-spn	Specify the service principal name (SPN) of the target computer. This option takes effect only when the –k or –K option is used. This option is equivalent to specifying the computer's service principal name for the **Service principal name** in the Delinea PuTTY Kerberos configuration page.

The other Kerberos settings—such as Create forwardable tickets and Find machine from trusted domains—are not exposed as options to the pscp.exe, psftp.exe and plink.exe programs. You can configure these settings using the Delinea PuTTY client user interface, save them in a session, then load the session using the –load option.

The following example illustrates how to use Delinea PuTTY command line options to facilitate administrative tasks. In this example, the pscp.exe program is used to retrieve the file /etc/group from a remote Linux computer named RedHatLinux with the current user's login name and Kerberos credentials for

authentication on the remote computer:

```
pscp –K RedHatLinux:/etc/group c:\temp
```

Because this command uses the -K option, you don't need to specify a user name in the command line or be prompted for password during runtime. Therefore, the command can be embedded in a batch file for administrative use. However, this command would require the remote RedHatLinux computer to have the Server Suite Agent installed and be joined to an Active Directory domain.

Getting More Information

For more information about the open-source version of PuTTY and standard PuTTY documentation, see the following resources:

- PuTTY website: http://www.chiark.greenend.org.uk/~sgtatham/putty/
- PuTTY documentation: http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html

Once you have installed and finished setting up your Delinea product and the RSA SecurID authentication agent, you can configure settings so that user authentication can occur for locally defined UNIX users or for Active Directory users who have UNIX profiles in the appropriate zone. In addition, specific groups of Active Directory users can be prompted for password authentication or two factor authentication.

The installation process for each agent does not interfere with or touch any configuration file used by the other product. Follow the standard installation steps for each product.

You can install the products in either order. After the DirectControl agent is installed, you need to join the computer to a domain and place it in a DirectControl Zone.

You can configure the Delinea Authentication Service and RSA SecurID to work together in either of two ways:

- Configure the PAM modules to work with the Authentication Service and RSA SecurID
- Configure SecurID for use with Server Suite zone-based role and privilege execution

If you're using an older version of authentication service or using a version that does not include multi-factor authentication (MFA) support, you can configure the PAM modules to work with authentication service and RSA SecurID. If you've configured role definitions or command rights to require MFA, you can rename a file and create a symlink to configure RSA SecurID to work with your authentication service deployment.

You need to have authentication service installed, with an agent on your UNIX/Linux computer.

You need to also have the RSA SecurID authentication agent installed and configured. This guide shows you how to configure the authentication service to prompt for a SecurID token.

RSA Installation Prerequisites

This guide assumes that you've already installed the RSA SecurID authentication agent. You can get more information about the RSA authentication agent at the following link:

https://www.rsa.com/en-us/products-services/identity-access-management/securid/authentication-agents/authentication-agents-for-pam

Installing the RSA Authentication agent includes but is not limited to the following tasks (consult the RSA documentation for a complete list):

- RSA Secure Console is set up for use
- In the RSA Secure Console, you've added your users, computers, and generated the sdconf.rec file.
- You've successfully installed the RSA authentication agent on your Linux and UNIX computers (this includes installing the sdconf.rec file).
- You've successfully tested the user authentication with the RSA acetest command.

If you have installed the RSA Authentication agent for PAM and successfully performed a test authentication for each user, then you're ready to configure DirectControl to work with the RSA agent and SecurID token.

The installation process for each agent does not interfere with or touch any configuration file used by the other product. Follow the standard installation steps for each product.

You can install the products in either order. After you install the Server Suite Agent, you need to join the computer to Active Directory and place it in a DirectControl Zone.

Installation Overview

To install and configure authentication service and RSA SecurID (an overview):

1. Install the DirectControl agent for *NIX.

   For details, see the Server Suite documentation.

2. Install and set up the RSA SecurID agent.

   For details, see the RSA document, "*RSA Authentication Agent 7.1 for PAM--Installation and Configuration Guide for RHEL*." The document is included in the agent download package.

3. Run the RSA acetest command to verify that the user login credentials work.

   For details, see the RSA documentation.

4. If you have configured role definitions or command rights to require multi-factor authentication (MFA), you create a symlink to point to the RSA SecurID authentication file instead of the file for DirectControl. For details, see Configuring SecurID for use with Server Suite zone-based role and privilege execution.

   With MFA enabled for role definitions or command right definitions, you don't have to manually configure each authentication module to use RSA SecurID.

5. If you use Authentication Service but you don't use role definitions or command right definitions configured for MFA:

   1. Modify the PAM authentication files for Linux, Solaris, or AIX:

      1. For Linux: Configure the /etc/pam.d/system-auth file:

         For details, see Configuring the /etc/pam.d/system-auth File for Linux.

      2. For Solaris and AIX: Configure the pam.conf file:

         For details, see Configuring the pam.conf File for Solaris and AIX.

   2. (Optional) Configure the system to use the SecurID for authentication for specific users or groups.

      Tip: It may be a good idea to disable SecurID authentication for the root user, at least initially, so that you don't get locked out of the computer entirely.

   3. (Optional, as needed) Configure SSH or other authentication services to use SecurID. For details on configuring SSH, see Configuring the pam.conf file for Solaris and AIX.

Configure the /etc/pam.d/system-auth File for Linux

After you've installed both the RSA SecurID and Server Suite Agents on a Linux computer, you'll also need to insert a line in the /etc/pam.d/system.auth file. This change will make it so that the system prompts users for their SecurID token.

Just so that you know, this file will already have some lines at the top that were inserted by the authentication service.

To configure the Linux system authentication file so that users are prompted for the RSA token:

- Add the following line to the beginning of the /etc/pam.d/system.auth file:

  auth required pam_securid.so

  You should restart any services that you plan to use with RSA. For example, if you're using SSH, you should restart the SSH service.

Configure the pam.conf File for Solaris and AIX

For Solaris and AIX computers, you need to edit the /etc/pam.conf file.

To configure the Solaris or AIX system authentication file so that users are prompted for the RSA token:

In the /etc/pam.conf file, add the following code snippet to the end of the file:

```
# Support for Kerberos V5 authentication and example configurations can
# be found in the pam_krb5(5) man page under the "EXAMPLES" section. sshd-kbdint auth required pam_securid.so
sshd-kbdint auth sufficient pam_centrifydc.so unix_cred
sshd-kbdint auth requisite pam_centrifydc.so deny sshd-kbdint account sufficient pam_centrifydc.so unix_cred
sshd-kbdint account requisite pam_centrifydc.so deny
sshd-kbdint session required pam_centrifydc.so
sshd-kbdint password sufficient pam_centrifydc.so ry_first_pass
sshd-kbdint auth requisite pam_authtok_get.so.1
sshd-kbdint auth required pam_dhkeys.so.1
sshd-kbdint auth required pam_unix_cred.so.1
sshd-kbdint auth required pam_unix_auth.so.1
sshd-kbdint account requisite pam_roles.so.1
sshd-kbdint account required pam_unix_account.so.1
sshd-kbdint session required pam_unix_session.so.1
sshd-kbdint password required pam_dhkeys.so.1
sshd-kbdint password requisite pam_authtok_get.so.1
sshd-kbdint password requisite pam_authtok_check.so.1
sshd-kbdint password required pam_authtok_store.so.1
```

You should restart any services that you plan to use with RSA. For example, if you're using SSH, you should restart the SSH service.

Require Token Authentication for Specific Groups or Local Users

RSA supports the ability to require RSA token authentication for specific groups of users. This feature is supported when using the Authentication Service. You can specify Active Directory groups as the required group. Local groups work as well.

You can also configure the agent so that specific groups are not prompted to authenticate with the RSA SecurID token. Group members excluded from SecurID authentication can authenticate using UNIX credentials or by way of another PAM module; you can configure this

Note: The ability to require RSA SecurID token authentication for specific groups does **not** work with AIX. There is a bug in the AIX OS that prevents the SecurID agent from iterating Active Directory groups.

Note: Be sure to exclude any users that you do not want to authenticate with the RSA SecurID token. Once you've enabled users or groups for token authentication, then all users will be challenged for a token even if they weren't assigned on. This situation can cause some users to be locked out of the computer that they're trying to log in to. When you are testing this functionality, it's a good practice to exclude the root user to avoid any complications.

To require SecurID token authentication for specific groups or users:

1. Edit the sd_pam.conf file and add the following lines:

   ```
   #VAR_ACE :: the location where the sdconf.rec, sdstatus.12 and securid files will go
   VAR_ACE=/opt/RSA
   ```

2. To specify specific groups to authenticate using the RSA token, first enable group support by setting the ENABLE_GROUP_SUPPORT parameter to 1, as shown below:

   ```
   #ENABLE_GROUP_SUPPORT :: 1 to enable; 0 to disable group support
   ENABLE_GROUP_SUPPORT=1
   ```

3. To specify the list of groups that will use the RSA token, include them in the LIST_OF_GROUPS parameter, as shown below:

   ```
   #LIST_OF_GROUPS :: a list of groups to include or exclude...Example
   #LIST_OF_GROUPS=other:wheel:eng:othergroupnames
   LIST_OF_GROUPS=sampleadgroup
   ```

4. To exclude groups from requiring the RSA token, include them in the INCL_EXCL_GROUPS parameter, as shown below:

   ```
   #INCL_EXCL_GROUPS :: 1 to always prompt the listed groups for securid
   # authentication (include)
   # :: 0 to never prompt the listed groups for securid
   # authentication (exclude) INCL_EXCL_GROUPS=1
   ```

5. (Optional) To configure what happens when an excluded user tries to authenticate, modify the PAM_IGNORE_SUPPORT parameter, as shown below:

   ```
   #PAM_IGNORE_SUPPORT :: 1 to return PAM_IGNORE if a user is not SecurID
   # authenticated due to their group membership
   # :: 0 to UNIX authenticate a user that is not SecurID
   # authenticated due to their group membership
   PAM_IGNORE_SUPPORT=1
   ```

6. To specify specific users to authenticate using the RSA token, first enable user support by setting the ENABLE_USERS_SUPPORT parameter to 1, as shown below:

   ```
   #ENABLE_USERS_SUPPORT :: 1 to enable; 0 to disable users support
   ENABLE_USERS_SUPPORT=1
   ```

7. To specify the list of users that will use the RSA token, include them in the LIST_OF_USERS parameter, as shown below:

   ```
   #LIST_OF_USERS :: a list of users to include or exclude...Example
   LIST_OF_USERS=localuser1:aduser2
   ```

8. To exclude users from requiring the RSA token, include them in the INCL_EXCL_USERS parameter, as shown below:

   ```
   #INCL_EXCL_USERS :: 1 to always prompt the listed users for securid
   # authentication (include)
   # :: 0 to never prompt the listed users for securid
   # authentication (exclude) INCL_EXCL_USERS=1
   ```

9. (Optional) To configure what happens when an excluded user tries to authenticate, modify the PAM_IGNORE_SUPPORT_FOR_USERS parameter.

You can also consult the RSA SecurID documentation for more details about configuring token authentication for groups, users, excluding users, and so forth. There are more configurations available than are presented in this document.

Configure SSH to Require SecurID

When setting up the SecurID product you must make some configuration changes to the sshd configuration files.

If you are using the Delinea openSSH product you must make some configuration changes to support token authentication. The Delinea openSSH is configured to attempt Kerberos single sign-on whenever a user logs in. This means that the user is not prompted for their user name or password. This capability must be disabled if you want to prompt users for token authentication.

To configure SSH to require a SecurID token:

1. Edit the /etc/centrifydc/ssh/ssh_config file and comment out the lines for the following items:
    - GSSAPIAuthentication
    - GSSAPIKeyExchange
    - GSSAPIDelegateCredentials

    For example:

    # Configuration for DirectControl: Host *
    #GSSAPIAuthentication yes
    #GSSAPIKeyExchange yes
    #GSSAPIDelegateCredentials yes
2. Edit the /etc/centrifydc/ssh/sshd_config file and comment out the lines for the following items:
    - GSSAPIKeyExchange
    - GSSAPIAuthentication
    - GSSAPICleanupCredentials
3. In the /etc/centrifydc/ssh/sshd_config file, be sure that the PrintMotd and UsePam settings are set as followings:

    PrintMotd no
    UsePAM yes
4. Restart sshd to ensure the changes take effect.

For the users that you want to use the SecurID pass code for login, you modify the affected role definitions to require multi-factor authentication. For the commands where you want users to provide a SecurID pass code, you configure the command right for re-authentication using multi-factor authentication.

To configure RSA SecurID for use with Server Suite zone-based role definitions and command rights:

1. In Access Manager, configure your role definitions to use multi-factor authentication:
    1. In Access Manager, locate the role definitions for which you want to require use of the SecurID pass code.

       For example, navigate to your zone, then go to **Authorization** › **Role Definitions**, and then select the rights definition in the right pane.
    2. For each role definition, right-click the role definition and select **Properties**.
    3. Click the **Authentication** tab.
    4. Select **Require multi-factor authentication for login**.
    5. Click **OK** to save the changes.
2. In Access Manager, configure your command rights to use multi-factor authentication:
    1. In Access Manager, locate the command rights definitions for which you want to require use of the SecurID pass code.

       For example, navigate to your zone, then go to **Authorization** › **UNIX Right Definitions** › **Commands**, and then select the rights definition in the right pane.
    2. For each command right, right-click the command right and select **Properties**.
    3. Click the **Attributes** tab.
    4. Select **Re-authenticate current user**.
    5. Select **Require multi-factor authentication**.
    6. Click **OK** to save the changes.
3. Make sure that you've installed the DirectControl agent for *NIX on the UNIX or Linux computer where you want users to use the RSA SecurID pass code.
4. On the Linux or UNIX computer where you want users to use the SecurID pass code, locate the pam_centrifydc_cloud.so file.
5. Rename the pam_centrifydc_cloud.so file.
6. Create a symlink for the pam_centrifydc_cloud.so file to point to the pam_securid.so file instead.

   For the affected users on the affected UNIX or Linux computers, those users will now need to enter their RSA SecurID pass code in order to log in to those computers.

To verify the authentication service and SecurID setup:

1. On the RSA Administration Server, add and configure a UNIX user.
2. Confirm that the local UNIX user can log in using the SecurID token by running the RSA acetest command.
3. In Access Manager, create a UNIX profile for a user in the zone where the UNIX machine is registered.
4. On the RSA Administration Server, register the UNIX profile user and assign them a SecurID token.
5. On the UNIX computer, log in with the new user.

   Tip: Use the UNIX login user name, not the Active Directory user name, when logging in to the UNIX computer.

If you need to disable a user's access to a particular computer, you can do so by one of three ways:

- Disable the user's Active Directory Account.
- Remove the user from the Server Suite Zone.
- Deselect the "Enable user access to this zone" option in the user's Centrify Profile tab.

- For sshd_config, you should explicitly set the following parameter to Yes. Even though the parameter is defaulted to this value, it sometimes is not correctly set. Without this parameter, you will not receive prompts for events like New Pin, and so forth.

  ChallengeResponseAuthentication Yes

- Even though the user authenticates with their SecurID token, they may be prompted to reset their Active Directory password if it has expired in the domain. After the user logs in, they will be presented with the “Change Password” prompts from Active Directory.
- When a user authenticates with a SecurID token, they are granted access to the UNIX machine, but they are not authenticated to the Active Directory Domain. As a result, they will not have Kerberos Credentials or single sign-on capability to other systems. After signing on, the user may type the following and then enter their Active Directory password to authenticate to Active Directory.

  >kinit

Server Suite centrally secures cross-platform data centers through Active Directory-based identity and access management for a wide range of heterogeneous systems, hypervisors and applications.

Built on an integrated architecture that leverages patented technology, the Server Suite of solutions help centralize ID, access privilege delegation and policy management to reduce the organization's IT expense and complexity, improve end-user productivity, strengthen security and enhance regulatory compliance initiatives. Key components of audit and monitoring service include integrated authentication, access control, role-based privilege management, user-level auditing and server protection solutions.

This book describes how to integrate the Samba open source file and print sharing program on a Linux or UNIX computer that has the DirectControl agent already installed.

> **Note**: Beginning in calendar year 2016, Centrify no longer supports the Centrify-enabled version of Samba that was available for use with earlier Server Suite releases. If you are currently using Centrify-enabled Samba with Server Suite 2013.3 or later, you must uninstall Centrify-enabled Samba, install open-source Samba, and install the latest version of the adbindproxy package. Those steps are described in Installing the Centrify Samba integration components. After you perform those steps, Server Suite (2013.3 or later) is integrated with open-source Samba.

Intended Audience

This book is written for an experienced system administrator familiar with the unpacking and installation of programs on Linux or UNIX computers. In addition, the instructions assume that you have a working knowledge of Samba and how to perform common administrative tasks for creating and maintaining Samba shares.

This book also requires you to have a working knowledge of Server Suite and how to perform common administrative tasks using the Access Manager console and the Active Directory Users and Computers administration tool. If you are unfamiliar with Server Suite, see the Administrator's Guide for Linux and UNIX and other documentation.

Using this Guide

The book guides you through the installation and configuration of the components necessary to integrate Server Suite and Samba. It is organized as follows:

- Using Server Suite Technology with Samba provides a brief overview of Samba, and how Samba, Centrify Authentication Service, and Active Directory work together to provide a secure, integrated environment.
- Installing the Centrify Samba Integration Components describes how to unpack and install the Centrify adbindproxy package.
- Migrating Existing Samba Users to Server Suite describes how to migrate your existing Samba users to Active Directory for use with Server Suite.
- Configuring the Samba Integration describes how to use the Samba configuration file and test your integration of Samba, Centrify Authentication Service, and Active Directory.
- Using adbindproxy.pl describes the adbindproxy.pl utility, which enables you to configure Samba for interoperability with Centrify Authentication Service.

Using Server Suite technology with Samba

These topics describe how Samba integrates with Server Suite, and highlights some integration issues that you might encounter.

What is Samba?

Samba is an open source file and printer sharing program that allows a Linux or UNIX host to participate as an Active Directory services domain member. When Samba is installed, Windows users can share files and printers on the Linux or UNIX computers.

Samba.org distributes the Samba files and expects users to download and build their own packages. All major Linux and free UNIX distributions have Samba as a native package. For a native install of Samba on your system, see your distributor's package or port system.

Also, the https://samba.plus web site offers Samba packages for Red Hat Enterprise Linux (RHEL), SuSE Linux Enterprise Server (SLES), and Debian systems. The http://en.opensuse.org/Samba web site offers Samba packages for all SuSE Linux products, including SLES.

What is Server Suite-enabled Samba?

Server Suite-enabled Samba is an adbindproxy module and PERL configuration script that enables Server Suite and Samba to work together without UID, GID, or Active Directory conflicts.

In previous releases, Server Suite would modify the Samba package and provide a unique, Server Suite version of Samba for different operating systems. In this release, Server Suite provides a couple of components that work with the stock Samba packages.

Server Suite is an integrated set of commercial identity management products that enable a Linux, UNIX, or Mac host to participate as an Active Directory domain member. When you install Server Suite products, you can manage the Server Suite-managed computer's user and group accounts and privileges entirely through Active Directory.

When open-source Samba is configured as an Active Directory domain member and the DirectControl agent is installed together with Samba on the same Linux or UNIX host, two problems can arise:

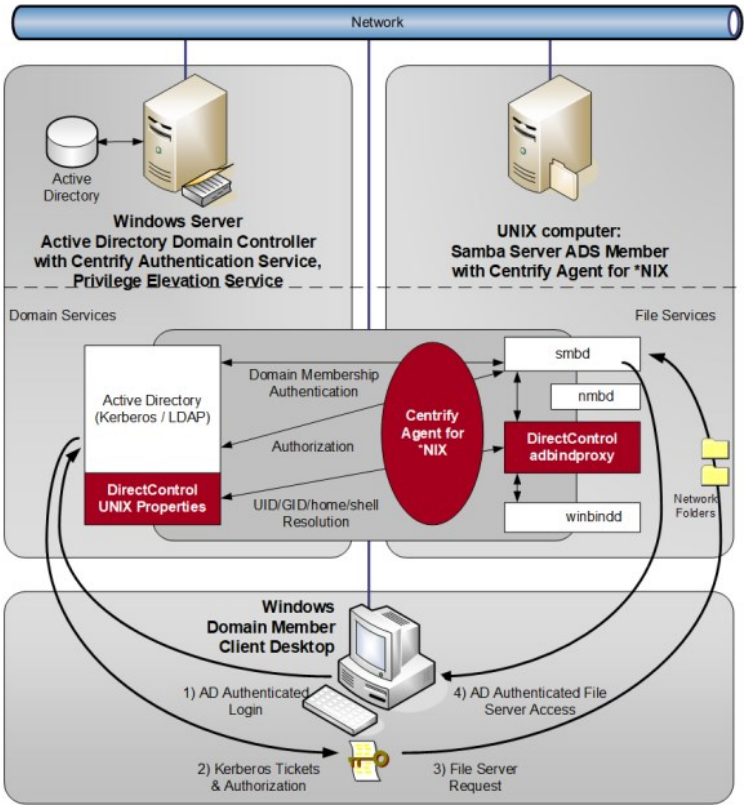
- Samba and the DirectControl agent both attempt to create and manage the same Active Directory computer account object, causing one of the products to stop working.
- Conflicting UIDs and GIDs are generated by Samba and the Server Suite Management Services tools for the same Active Directory users and groups. However, the two programs use different algorithms for generating these values. The result is file ownership conflicts and access control problems.

To resolve these issues, Server Suite provides the following components:

- **adbindproxy (adbindd) module:** The adbindproxy module uses the adbindd daemon. Unless otherwise noted, "adbindproxy" and "adbindd" are used interchangeably in the documentation. The adbindproxy (adbindd) module intercepts Samba UNIX ID mapping requests and reroutes them to the DirectControl agent for processing. This module ensures that Samba and DirectControl agent agree on the UNIX attribute values.
- **adbindproxy.pl PERL configuration script:** Automates most of the setup process and designates the DirectControl agent as the manager of the shared computer object.

Server Suite-Enabled Samba Architecture

The following figure provides a conceptual view of the complete solution architecture using Active Directory, Samba, and Server Suite for Samba components.



If you have not been using Samba up to this point, or if you have been using an older Samba security method (such as user or server), the integration process makes it easy to configure Samba as an Active Directory member.

On the other hand, if you have already been using Samba as an Active Directory domain member and have assigned UIDs and GIDs to Active Directory users and groups, the PERL configuration script helps to resolve conflicts when Samba and Server Suite are integrated.

The integrated solution, composed of the DirectControl agent (installed separately), open-source Samba, and adbindproxy, provides the following:

- Samba and the DirectControl agent use the same Active Directory computer object without conflicts.
- Consistent user and group attributes are applied on files across Windows, Linux and UNIX computers.
- All UNIX user identity attributes, including the UID, GID, home directory, and login shell in UNIX profiles, are centrally stored and managed in Active Directory.
- Both Kerberos and NTLM Samba authentication methods are supported.
- Standard Samba access-control features are implemented and augmented by the Server Suite zones technology.

Installing the Centrify Samba Integration Components

This section explains how to install the Server Suite adbindproxy package. You install the adbindproxy package on your Linux and UNIX computers so that the DirectControl agent works with Samba.

Installation Process Overview

Your Linux or UNIX computer can be in one of three main states regarding Samba and Server Suite:

- New to both Server Suite and Samba:

  Samba is not in use and the computer does not have the DirectControl agent installed. The Samba packages might already be installed but you haven't configured Samba yet. For details, see Installation Overview for Computers New to both Server Suite and Samba.

- Using Samba, new to Server Suite:

  Samba is in use but the computer doesn't have the DirectControl agent installed. For details, see Installation Overview for Computers New to Server Suite.

- Using the previous Centrify-enabled version of Samba:

  Samba is in use and the DirectControl agent is installed, and you're using the previous release of Centrify-enabled Samba. For details, see Upgrade overview for Computers with Centrify-Enabled Samba.

The installation process varies slightly depending on what kind of environment you're currently using.

Installation Overview for Computers New to both Server Suite and Samba

If you're configuring a computer that does not yet have either Samba working nor the DirectControl agent, here's an overview of what you need to do.

Make sure that you have the software you need.	Make sure that you have the latest version of the DirectControl agent, the Centrify adbindproxy package, and the open source Samba files.
Install the DirectControl agent.	Refer to the Server Suite documentation for instructions.
Install open source Samba.	All major UNIX and Linux distributions have Samba as a native package. See your distributor's package or port system for a native install of Samba on your system. You can also visit https://samba.plus/ which offers Samba packages for Red Hat Linux, SUSE Linux Enterprise Server, and Debian.
Install the Server Suite adbindproxy package.	See Installing the adbindproxy Components
Run the adbindproxy.pl script.	See Configuring the Samba Integration
Modify the Samba configuration file, as needed.	See Modifying the Samba smb.conf Configuration File.
Test and verify the configuration.	See Verifying the Samba Integration

Installation Overview for Computers New to Server Suite

If you're configuring a computer that has Samba configured but that does not yet have the DirectControl agent installed, here's an overview of what you need to do.

Make sure that you have the software you need.	Make sure that you have the latest version of the DirectControl agent, the Centrify adbindproxy package, and the open source Samba files.
Install the DirectControl agent.	Refer to the Server Suite documentation for instructions.
Make a backup copy of your smb.conf file.	
Install the Centrify adbindproxy package.	See Installing the adbindproxy Components
Migrate Samba users to Active Directory.	See Migrating Existing Samba Users to Server Suite **Note**: If you're using Auto Zone or Server Suite Express, user migration is not supported.
Run the adbindproxy.pl script.	See Configuring the Samba Integration
Modify the Samba configuration file, as needed.	See Modifying the Samba smb.conf Configuration File.
Test and verify the configuration.	See Verifying the Samba Integration

Upgrade Overview for Computers with Server Suite-Enabled Samba

Beginning in calendar year 2016, we neither provide nor support the Server Suite-enabled version of Samba that was available earlier. Instead, we now provide a standalone adbindproxy package containing the components that are necessary for Server Suite to integrate with open-source Samba.

If you are currently using Server Suite-enabled Samba with Server Suite 2013.3 or later (Server Suite), not only do you need to upgrade to the latest DirectControl agent but there are some additional steps to migrate your users and settings. Below is an overview of what you need to do on each agent-controlled Linux and UNIX computer that was integrated with Samba.

Make sure that you have the software you need.	Make sure that you have the latest version of the DirectControl agent, the Centrify adbindproxy package, and the open source Samba files.
Make a backup copy of your smb.conf file.	
Uninstall Server Suite-enabled Samba.	For example, on most Linux variants you would issue the following command: `rpm -e CentrifyDC-samba`
Upgrade the DirectControl agent so that it's either the latest version or a version later than 2013.3.	Refer to the Server Suite documentation for instructions.
Install open source Samba.	All major UNIX and Linux distributions have Samba as a native package. See your distributor's package or port system for a native install of Samba on your system. You can also visit https://samba.plus/ which offers Samba packages for Red Hat Linux, SUSE Linux Enterprise Server, and Debian.
Install the Server Suite adbindproxy package.	See Installing the adbindproxy Components

Migrate Samba users to Active Directory.	See Migrating Existing Samba Users to Server Suite **Note**: If you're using Auto Zone or Server Suite Express, user migration is not supported.
Run the adbindproxy.pl script.	See Configuring the Samba Integration
Modify the Samba configuration file, as needed.	See Modifying the Samba smb.conf Configuration File.
Test and verify the configuration.	See Verifying the Samba Integration.

What's in the adbindproxy Package

After you download and extract the Centrify adbindproxy package, you'll see the following files: ./Centrify-Adbindproxy-Release-Notes.html ./CentrifyDC-adbindproxy-*release*-rhel5-x86_64.rpm

The software bundle has a name in this format: centrify-adbindproxy-*release*-rhel5-x86_64.rpm and it contains these components:

- **adbindproxy (adbindd) module:** The adbindproxy module uses the adbindd daemon. Unless otherwise noted, adbindproxy and adbindd are used interchangeably in the documentation. The adbindproxy (adbindd) module intercepts Samba UNIX ID mapping requests and reroutes them to the DirectControl agent for processing. This module ensures that Samba and the DirectControl agent agree on the UNIX attribute values.
- **adbindproxy.pl PERL configuration script**: This script automates most of the setup process and designates the DirectControl agent as the manager of the shared computer object.

Installing the adbindproxy Components

Perform the following steps to install the integration components from the adbindproxy package. In these steps, the file name CentrifyDC-adbindproxy-*.rpm is used in place of the full file name. You can use the wildcard symbol (*) to substitute for a portion of the file name if there are no conflicting files in the directory.

> **Note**: If you are upgrading from a previous version of Server Suite-enabled Samba, see Upgrade overview for computers with Centrify-enabled Samba before proceeding.

Be sure to enter the full path name in the command line if multiple versions of the same file exist in the same directory.

To install the Centrify Samba integration components

1. Run the appropriate command for your platform to install the centrifydc-adbindproxy package.

   The following table shows sample commands using the common package installers for each platforms.

Linux-based computers Red Hat Enterprise Linux	For 64-bit systems: rpm -Uvh CentrifyDC-adbindproxy-*release*-rhel5.x86_64.rpm For PowerPC systems: rpm -Uvh CentrifyDC-adbindproxy-*release*-rhel5.ppc64.rpm For Little-endian PowerPC systems (PPCLE): rpm -Uvh CentrifyDC-adbindproxy-*release*-rhel7.ppc64le.rpm
Sun Solaris	On SPARC systems, for example: gunzip centrifydc-adbindproxy-*release*-sol10-sparc-local.tgz tar -xf centrifydc-adbindproxy-*release*-sol10-sparc-local.tar pkgadd -d CentrifyDC-adbindproxy For other Solaris versions and platforms, the commands are the same but the filenames are different. For example, on a 64-bit system: centrifydc-adbindproxy-*release*-sol10-x86-local.tgz
	For HP-UX 11.31 on PA-RISC: gunzip centrifydc-adbindproxy-*release*-hp11.31-pa.depot.gz

HP-UX	swinstall -s /path/centrifydc-adbindproxy-*release*-hp11.31-pa.depot CentrifyDC-adbindproxy For other HP-UX versions and platforms the commands are the same but the file names are different. For example on HP-UX 11.31 Itanium 64-bit systems: centrifydc-adbindproxy-*release*-hp11.31-ia64.depot.gz
IBM AIX	For AIX 7.1 or later: gunzip centrifydc-adbindproxy-*release*-aix7.1-ppc-bff.gz inutoc installp -aY -d centrifydc-adbindproxy-*release*-aix7.1-ppc-bff CentrifyDC.adbindproxy
Debian Linux Ubuntu Linux	Check that you have libcupsys2-gnutls10 (1.1.23-1 or later) installed If you have the required libraries, run the following command to install: dpkg –i centrifydc-adbindproxy-*release*-deb8-x86_64.deb
SuSE Linux OpenSuSE Linux	For 64-bit systems: rpm -ivh CentrifyDC-adbindproxy-*release*-suse11.x86_64.rpm

2. (Optional) Join the computer to a zone using the adjoin command.

   This concludes the installation of the adbindproxy package.

   If you have existing Samba users to migrate, go to Migrating Existing Samba Users to Server Suite. Otherwise, go to Configuring the Samba Integration to continue.

Updating the Samba Files

After you've installed the Server Suite adbindproxy package, you might need to update your version of Samba. When you update the Samba files, the update will replace smb.conf and also restart Samba with its own startup script instead of the adbindd script.

Before you update your version of Samba, it's a good practice to make a backup copy of your smb.conf file.

After you update your version of Samba, perform the following tasks so that you can keep the Server Suite adbindproxy package working.

To keep the Server Suite adbindproxy package working after updating Samba:

- Do one of the following:
  - Run adbindproxy.pl to reconfigure the centrifydc-samba service (Recommended)

    After adbindproxy.pl finishes the setup, you may want to add back the customized settings from the smb.conf backup to the new smb.conf file. Restart the centrifydc-samba service after the change. Note that the commands to restart the service are different on different platforms.
  - Manually replace the smb.conf with the backup.

    After replacing the smb.conf file, restart the centrifydc-samba service. Note that the commands to restart the service are different on different platforms.

    This method may not work because the Samba upgrade may affect the configurations of the centrifydc-samba service and the Samba service itself.

Migrating Existing Samba Users to Server Suite

This section describes how to migrate an existing user population from Samba servers to the integrated Server Suite.

> **Note**: The information in this section is relevant to computers with the core Server Suite components installed and for which you created a Server Suite zone. These instructions do not apply to computers with Server Suite Express installed or computers that are joined through Auto Zone. If you are using Server Suite Express or if you have joined a computer using workstation mode, it is not possible to migrate existing Samba UID and GID settings.

Migrating UNIX Profiles to Active Directory

If your current environment includes Samba servers that are joined to the Active Directory domain as member servers and existing Windows users access the data on those servers, you may want to migrate those existing users to Server Suite to rationalize UIDs and GIDs and manage all of your network's conflicting identities in a single, centralized ID repository.

> **Note**: Migrate your Samba users to Active Directory, as explained in this section, **before** integrating Samba and Authentication Service as explained in Running the adbindproxy.pl Script.

There are two ways to migrate your UNIX profiles to Active Directory:

- If winbind is currently configured in your /etc/nsswitch.conf file, you need to run the getent command to retrieve the user information.
- If you do not have winbind configured in your /etc/nsswitch.conf file, then run the adbindproxy perl script to migrate the users. See the instructions below.

Migrating Users if Winbind is Configured in /etc/nsswitch.conf

To save the winbind information to a file:

1. If winbind is currently configured in your /etc/nsswitch.conf file, run the following commands to save the information to a file before installing the adbindproxy package:

   ```
   getent passwd | grep -v -f /etc/passwd > /tmp/passwd.winbind
   ```

   ```
   getent group | grep -v -f /etc/group > /tmp/group.winbind
   ```

2. Move the exported files to a computer where you have installed the Access Manager console.
3. In the Access Manager console, use the **Import from UNIX** wizard to import the users and groups (with their existing UID and GID mappings) into the zone.

   For more information on importing existing user and group information and mapping information to Active Directory, see the "Importing existing users and groups" chapter in the *Administrator's Guide for Linux and UNIX*.

Migrating Users with the adbindproxy perl Script

If winbind is not currently configured in your /etc/nsswitch.conf file, follow the steps below after you've installed the adbindproxy package.

This script gets the UID andGID files from Samba. You then import them into Active Directory.

To migrate UNIX user profiles to Active Directory using the adbindproxy.pl script:

1. Identify the Samba servers you want to update to integrate with Server Suite.
2. On each of the Samba servers to be updated, locate the winbindd_idmap.tdb file and create a backup copy of the file.
   1. To locate the winbindd_idmap.tdb file, you can run a command similar to the following to view details about the Samba build:

      ```
      /CurrentSambaBInaryPath/smbd -b |grep -i lockdir
      ```

   2. In the output, you should see a line similar to the following that indicates the location of the winbind_idmap.tdb file:

      ```
      LOCKDIR: /var/lib/samba
      ```

3. Make a backup copy of the winbindd_idmap.tdb file.

   For example:

```
cp /var/lib/samba/winbind_idmap.tdb /tmp/winbind_idmap.tdb.pre_adbindproxybackup
```

4. Run the adbindproxy.pl script with the following options to generate the export files.

   ```
   perl /usr/share/centrifydc/bin/adbindproxy.pl --export --groupFile filename --userFile filename --tdbFile filename
   ```

   See Using adbindproxy.pl for details about the command-line parameters for adbindproxy.pl.

   When you run these adbindproxy.pl options it generates export files for the users and the groups that are currently known by the Samba server. By default, these files are created as:

   ```
   /var/centrify/samba/passwd
   ```

   ```
   /var/centrify/samba/group
   ```

5. Move the exported files to a computer where you have installed the Access Manager console.
6. In the Access Manager console, use the **Import from UNIX** wizard to import the users and groups (with their existing UID and GID mappings) into the zone.

   For more information on importing existing user and group information and mapping information to Active Directory, see the "Importing existing users and groups" chapter in the *Administrator's Guide for Linux and UNIX*.

Migrating Samba Servers to Server Suite Zones

Samba generates UIDs and GIDs based on a range of values that have been defined for a specific server. In most cases, a user who has accessed two different Samba servers is likely to have two different UIDs: for example, a user could have UID 6003 on the server mission and UID 9778 on the server dolores.

Therefore, in an initial migration of existing users, each Samba server must join the Active Directory domain in separate Server Suite Zones to accommodate the different UIDs and GIDs users and groups may have.

If you want users to have consistent GIDs and UIDs, then you need to put the Samba servers in the same zone.

Configuring the Samba integration

This section describes how to configure the DirectControl agent and Samba to work together properly after you have installed the integration components from the adbindproxy package and joined agent-controlled computers to a zone.

Running the adbindproxy.pl Script

This section describes how to configure Samba using the adbindproxy.pl script.

> **Note**: If your current environment has Windows users accessing data on Samba member servers that are joined to the Active Directory domain, you may want to migrate those users to Server Suite. This way, you can use Server Suite zones to manage conflicting identities and rationalize UIDs and GIDs. For details on how to migrate those users, see Migrating existing Samba users to Server Suite Complete the migration **before** integrating Samba and the Authentication Service.

The adbindproxy.pl script performs the following tasks:

- Determines the computer's operating system and adjusts accordingly.
- Confirms that the DirectControl agent is installed.
- Confirms that open-source Samba has been installed.
- Determines if you are joined to an Active Directory domain and, if you are, displays the domain name and Server Suite Zone.
- Asks if you want to join Samba to the current Active Directory domain or another. If you choose another, the script guides you through the current domain leave and new domain join processes.

  > **Note**: If you want to modify or set advanced join settings (for example, update PAM or NSS config, use DES for encryption, or use a computer alias), either run adleave before you run adbindproxy.pl or select a different domain when prompted in the script. Otherwise, the script does NOT prompt you to enter advanced join settings.

- If you have a previous Samba installation, asks if you want to keep the smb.conf settings or use new ones. adbindproxy.pl automatically saves the existing copy.

  > **Note**: The script automatically looks for an existing smb.conf file using the smbd -b command. If your current version of smbd does not support the -b option or you have smb.conf in a custom directory the script will not find it. If you want to use your existing smb.conf, move it to /etc/samba before you run the script.

- Removes old state files from previous instances of Samba, including any existing winbind entries from the /etc/nsswitch.conf file.
- Restarts the necessary clients (nmbd, winbindd, adbindd and smbd).
- Installs scripts to automatically start the correct Samba and Server Suite services each time the computer boots.
- Optionally can pass additional options for adjoin and adleave.
- Can generate a response file so that you can run the adbindproxy.pl script without any user interaction.

Before you run adbindproxy.pl, read through the prompts described below to make sure you're prepared with the answers. For example, before you run the script be sure you know the path where Samba is installed.

To begin, log on and switch to the root user and proceed with the following steps:

To run the adbindproxy.pl script

1. To start the script, from root enter the following:

   ```
   perl /usr/share/centrifydc/bin/adbindproxy.pl
   ```

2. Specify the path to the Samba installation:

   1. If Samba is not installed in the default location (/usr), enter the Samba path.
   2. If Samba is installed in /usr, press **Enter** to accept the default. Otherwise, enter your path.

3. Specify the domain to join.

   You proceed based on whether the computer is already joined to a domain or not:

   - If you **are already joined** to a domain when you initiated the script, the script displays the domain name and zone and asks you the following:

     ```
     Do you want to leave or join to another domain? [N]
     ```

To continue to join the current joined Active Directory domain press Enter and skip ahead to Step 6.

If you want to leave the current domain and join another OR change any advanced options (see the list below) in your current domain enter Y and then continue to Step 4.

- If you **are not joined** to a domain, the script displays the following message:

  ```
  Not joined to any domain. Make sure you enter the correct domain and zone information in the next steps
  ```

  This initiates a set of prompts that ask you for the Active Directory domain name, the Centrify Zone and advanced options.

  Continue to Step 4.

4. Join the new Active Directory domain.

   You arrive at this step if you are not joined to an Active Directory Domain when you started adbindproxy.pl or if you decided to leave that domain OR you decided to change advanced options in your current join. If none of these conditions apply to you, skip to Step 6.

   1. At this prompt, enter the domain name:

      ```
      Enter the Active Directory domain to join:
      ```

   2. At the DNS health prompt, press **Enter** to verify that the domain exists.

      ```
      Check DNS health for [domain]? Note: this may take several minutes [Y]:
      ```

   3. At the next prompt, enter the following domain properties:

      Note: If you are running Server Suite in Express Mode or need to join the domain through Auto Zone, enter NULL_AUTO for the zone name.

      1. Server Suite zone on the target Active Directory domain
      2. Computer name on which the adbindproxy package is installed
      3. Active Directory authorized user (default is Administrator)

5. (Optional) Specify advanced join options.

   The script prompts you with the following message:

   ```
   Do you wish to specify advanced join options? [N]:
   ```

   The options are listed below. The defaults are in brackets.

   1. If do not need any advanced join options, enter N. Otherwise, enter Y and make your selections.

      ```
      Canonical name of Active Directory Computer Container
      Preferred Domain Server to use (press Enter for none)
      Update PAM and NSS Config [Y]
      Trust computer for delegation? [N]
      Use DES encryption only? [N]
      Run adjoin in verbose mode? [N]
      Addition computer alias (press Enter for none)
      ```

      The script then displays the selections you made and asks if you want to proceed.

   2. Enter Y to proceed or N to abort adbindproxy.pl.

      If you were not joined to an Active Directory domain when you started the script, you are prompted to enter your password once.

   3. Enter the password for the Active Directory Domain, computer and authorized user specified in the prompts.

      Note: If you choose to proceed **AND** you are leaving the current Active Directory domain to join another, the script prompts you **twice** to enter your password.

   4. In response to the first prompt, enter the current Active Directory domain account password to leave that domain.

   5. In response to the second prompt, enter the password for the Active Directory Domain, computer and authorized user specified in the prompts to join the new domain.

6. Enter the Samba winbindd path.

   At the next prompt, if the samba winbindd listen path is not in /run/samba/winbindd, enter the path or press **Enter** to accept the default.

7. If there is an existing smb.conf file, continue to Step 8.

   Otherwise, if there is no existing smb.conf file (which is true for new installations of Samba), the adbindproxy script searches for existing smb.conf files. If it **does not** find an existing smb.conf file, it automatically creates a new one, stores it in /etc/samba, and displays the following message:

   ```
   Updating smb.conf with recommended settings ...
   ```

   and finishes the script.

   This new smb.conf file has minimal global settings and a samba-test share.

   > **Note**: Regardless of whether you update an existing smb.conf or create a new one, you will need to modify the /etc/samba/smb.conf file to have the [global] section settings and the appropriate shares for your environment. See Modifying the Samba smb.conf configuration file for instructions. The file created by adbindproxy.pl should be used for verifying the Samba integration only.

If you do have at least one existing smb.conf file, continue to Step 8.

8. Specify existing or new smb.conf settings:

   If you have an existing smb.conf file, you next specify whether to update the settings in the existing smb.conf file or create a new, skeletal smb.conf file. If you choose to use the existing settings, you can also choose to do a backup of the existing smb.conf file.

   If the script **does** find an existing smb.conf file, the script copies the smb.conf file to /etc/samba and asks the following question:

   ```
   Do you want to keep the original samba settings? [Y]:
   ```

   > **Note**: If the script finds more than one smb.conf, it displays the list and asks you to select one. After you make the selection, it copies that one to /etc/samba and continues.

   > **Note**: Regardless of whether you update an existing smb.conf or create a new one, you will need to modify the /etc/samba/smb.conf file to have the [global] section settings and the appropriate shares for your environment. See Modifying the Samba smb.conf configuration file for instructions. The file created by adbindproxy.pl should be used for verifying the Samba integration only.

- **Don't keep the original Samba settings**: Enter N to not keep the original Samba settings and instead create the new, basic smb.conf.

  The script creates a backup copy of your smb.conf in /etc/samba. The backup filename is in this format: smb.conf.yyyy-mm-dd-hh-mm. This new smb.conf file has minimal global settings and a samba-test share, if no shares exist.

  Continue to Finishing Up.

- **Keep the original Samba settings**: Enter Y to modify the existing file and continue to Step 9.

9. If you've chosen to keep the original Samba settings, the script displays the following prompt about backing up the existing settings:

   ```
   Backup existing /etc/samba/smb.conf and add recommended settings? [Y]
   ```

   - Enter Y to create a backup in the form, smb.conf.yyyy-mm-dd-hh-mm.
   - Enter N to use the existing smb.conf without making a backup.

     > **Note**: If the existing smb.conf has Security = ADS and the workgroup and realm are set, the script does NOT modify the existing file; the original is left unchanged.

10. For ubuntu and Suse computers where AppArmor exists, the script displays the following prompt about updating the AppArmor policy profiles:

    ```
    Update AppArmor policy profiles? [Y]
    ```

    Use the default [Y], unless you don't want to update the AppArmor profiles now.

    If you don't update the AppArmor profiles now, be sure to update them manually later. Otherwise, winbindd might fail to start and you won't be able to access the samba share. For ubuntu systems, the profiles aren't updated because the winbind policy profile doesn't exist.

11. If you're configuring a Linux system that has SELinux enabled and Samba supports your system's version of samba_selinux, the script checks the

configurations and, if needed, displays the following prompt:

```
Update SELinux policy to allow r/w on non samba_share_t types? [Y]
```

Use the default [Y] unless you have labeled all the share folders with the type samba_share_t.

If you don't update the SELinux policy, Samba cannot read or write to the shared folder is not labeled with the samba_share_t type.

For more information about samba_selinux, see the samba_selinux man page.

12. If you've chosen to keep the original Samba settings, the script displays the following prompt about resetting the Samba cache for user and group IDs.

    ```
    Reset the Samba User/Group ID Cache (Server Suite Samba may create conflicting mappings) [Y]
    ```

    Unless you have created custom mappings, use the default [Y]. This flushes the cache and displays the following message:

    ```
    This prompt is only pertinent to the small set of Samba administrators who created custom user and group ID mappings. If you do have custom mappings, use the default to flush the cache and prevent potential conflicts. After adbindproxy.pl completes, re-add your mappings as necessary.
    ```

    If you entered Y, the script creates new mappings in the Samba User/Group ID cache, which may result in conflicts if there are any mappings in place already.

Finishing Up

To complete the configuration, adbindproxy.pl stops any running versions of smbd, adbindd, winbindd and nmbd, starts the required Server Suite processes, and displays a set of progress and configuration messages. You should see the following messages:

```
Init Samba start script ...
Restarting Samba daemons ...
Reloading systemd: [ OK ]
Restarting centrifydc-samba (via systemctl): [ OK ]
Current DirectControl Configuration:
...
Current Samba Configuration:
...
```

The adbindproxy script displays the following:

```
Press ENTER to continue ...
Notes: If you need to join another domain, please re-run this script and enter the new domain name!
Done.
```

Note: If any service fails to start, you should run one of the following after the adbindproxy.pl script completes its execution.

On Linux or Solaris computers, run:

```
/etc/init.d/centrifydc-samba restart
```

On HP-UX computers, run:

```
/sbin/init.d/centrifydc-samba restart
```

On AIX computers, run:

```
stopsrc -g samba && startsrc -g samba
```

On Linux computers that support systemd, run:

```
systemctl restart centrifydc-samba
```

As a quick test, log off as the root user and log on with an Active Directory user account that has been granted access to the local computer's zone. If this is the first time that you are logging on with this user account, check that the user's home directory is created, which is created automatically by the Authentication Service the first time you log on.

Verifying the Samba Integration

To verify that Samba and Server Suite are working together correctly, you test if you can access Samba shares. If you upgraded existing shares, then you can test those; otherwise, you can verify the connection using the test share.

There are two key scenarios for testing whether Samba is configured properly for integration with the Authentication Service and Active Directory:

- Accessing Samba from a UNIX Client Session
- Accessing Samba Shares from a Windows Desktop

Accessing Samba from a UNIX Client Session

To test access to Samba shares on a Linux or UNIX computer, users should do the following:

To access Samba from a UNIX client session:

1. Log on to the Linux or UNIX computer using the Active Directory account that has been granted access to the local computer's zone.
2. Run the following command:

   ```
   smbclient -k -L host_name
   ```

   The smbclient program displays information about Samba and the SMB shares that are available on the local computer. For example, you should see a listing similar to the following (where s.s.s is the Samba version):

```
OS=[Unix] Server=[Samba s.s.s]

 Sharename Type Comment
 --------- ---- -------
 samba-test Disk
 IPC\$ IPC IPC Service (Samba-CDC)
 sara Disk Home directories

OS=[Unix] Server=[Samba s.s.s]

 Server Comment
 --------- -------
 Workgroup Master
 -------- -------
 ARCADE MAGNOLIA
```

If you are able to see the Samba shares as an Active Directory user logged on to the Linux or UNIX computer that is acting as the Samba server, you should next test accessing the Samba shares from a Windows desktop. For information about performing this test, see Accessing Samba shares from a Windows desktop.

Purging and Reissuing Kerberos Tickets on UNIX Computers

If you see an error such as NT_STATUS_LOGIN_FAILURE instead of the expected results when you run the smbclient program, you may need to purge your existing Kerberos tickets and have them reissued. Try running the following command to remove all of your Kerberos tickets:

```
/usr/share/centrifydc/kerberos/bin/kdestroy
```

Then run the following command to reissue tickets after you provide your Active Directory password:

```
/usr/share/centrifydc/kerberos/bin/kinit
```

You can then run the following command to list the Kerberos tickets that have been issued to you:

```
/usr/share/centrifydc/kerberos/bin/klist
```

After verifying the Kerberos tickets you have been issued, try running the smbclient program again.

Verifying the Version of Samba You Are Using

If purging and reissuing tickets does not resolve the problem, confirm the version of the smbstatus that is currently running using the following command:

```
smbstatus | grep version
```

The command should display the Samba version you have installed. For example:

Samba version s.s.s

(where s.s.s is the installed Samba version)

If the correct version of Samba is installed, run smbstatus again and note the names of any *.tdb files that do not exist, and try restoring them from your backup, then try running the smbclient program again.

If You Don't See the Correct Samba Shares

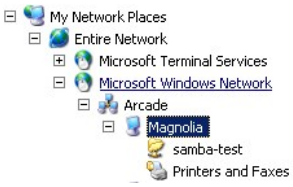
If the smbclient program does not display the Samba shares you have defined in the configuration file, you should review the settings in the smb.conf file and then restart the DirectControl agent and run the adflush command.

Accessing Samba Shares from a Windows Desktop

To test access to Samba shares on a Linux or UNIX computer from a Windows desktop:

1. Log on to a Windows computer that is joined to the domain with an Active Directory user account.
2. Click **Start › Windows Explorer**, then navigate to the domain.

   For example, open **My Network Places › Entire Network › Microsoft Windows Network › Arcade** to view the Arcade.net domain.
3. Select the Linux or UNIX computer that is integrated with Samba to view its Samba shares. For example:



4. Click samba-test or browse other available Samba shares to verify that you can open existing files and create new files.
5. Confirm from both Windows and the managed computer that the files in the share directories are owned by the correct users.

If you cannot browse the shares on the Linux or UNIX computer from the Windows desktop, you should:

- Verify that there is network connectivity between the two systems.
- Confirm that you do not have a firewall running on the managed computer that is blocking access to the SMB ports.
- Make sure there are no stale Kerberos tickets on your Windows system. The tools to remove stale Kerberos tickets may already be installed on your system—see this site for more information about klist and kerbtray programs.

Modifying the Samba smb.conf Configuration File

The Samba configuration file, /etc/samba/smb.conf, defines important parameters for Samba-based file sharing. After you have verified the Samba integration with the Authentication Service and Active Directory using a sample configuration file and the test share, you need to modify the smb.conf file so that it accurately represents your environment.

This smb.conf file must include the [global] section that defines the Active Directory domain, authentication methods, and other parameters. The file should also include a section for each directory you are making accessible as a SMB share.

At the beginning of a line, both the hash symbol (#) and the semi-colon (;) indicate lines to ignore. By convention, in this file, the hash indicates a comment and the semi-colon indicates a parameter you may wish to enable.

If you specify multiple users in valid users, user names can be separated by a comma or by white space.

The settings in the [global] section are required whether you use the sample configuration file or create your own smb.conf file. The settings in the [homes] section indicate that you want to share home directories, and the [samba-test] section describes the samba-test share as a publicly-writable share mapped to the /samba-test directory. For more information about editing the Samba configuration file and the supported parameters, see the Samba documentation.

A sample Samba smb.conf Configuration File

The adbindproxy script tests to determine what operating system is running on the host and generates an smb.conf file appropriate to that platform.

In the following sample file, it runs on a CentOS computer in the arcade.net domain and the Samba share is called MyShare.

```
#
# This file was generated by ADBindProxy Utility
#
[global]
    security = ADS
    realm = ARCADE.NET
    workgroup = ARCADE
    netbios name = centos-6
    auth methods = guest, sam, winbind, ntdomain
    machine password timeout = 0
    passdb backend = tdbsam:/var/lib/samba/private/passdb.tdb
    #
    # Samba versions 3.4.0 and newer have replaced "use kerberos keytab"
    # with "kerberos method".  The directive "kerberos method = secrets and keytab"
    # enables Samba to honor service tickets that are still valid but were
    # created before the Samba server's password was changed.
    #
    kerberos method = secrets and keytab
    #
    # Setting "client use spnego principal" to true instructs SMB client to
    # trust the service principal name returned by the SMB server. Otherwise,
    # client cannot be authenticated via Kerberos by the server in a different
    # domain even though the two domains are mutually trusted.
    #
   #  client use spnego principal = true
    #
    # Setting send spnego principal to yes .
    # Otherwise, it will not send this principal between Samba and Windows 2008
    #
   #  send spnego principal = Yes
    # If your Samba server only serves to Windows systems, try server signing = mandatory.
    server signing = auto
    client ntlmv2 auth = yes
    client use spnego = yes
    template shell = /bin/bash
    winbind use default domain = Yes
    winbind enum users = No
    winbind enum groups = No
    winbind nested groups = Yes
    idmap cache time = 0
   #  ignore syssetgroups error = No
    idmap config * : backend  = tdb
    idmap config * : range = 1000 - 200000000
    idmap config * : base_tdb = 0
    enable core files = false
    #  Disable Logging to syslog, and only write log to Samba standard log files.
    #syslog = 0
[samba-test]
    path = /samba-test
    public = yes
    #  if set  public = No, we should  set parameter valid users .
    #  and when the user or group is in AD , the setting syntaxes is:
    #  valid users = CPUBS\username +CPUBS\group
```

```
   writable = yes
[MyShare]
  path = /samba-test
  browsable = yes
  writable = yes
  guest ok = yes
  read only = no
[homes]
   comment = Home directories
   read only = No
   browseable = No
```

SMB.conf File Variations for Different Platforms

Some platforms will have slight variations in the smb.conf file, as follows:

- On HP-UX computers, the following line is added:

  ```
  guest account = smbnull
  ```

- On SuSE computers, the following lines are added:

```
# Suse 11 CUPS printing appears to crash at start up
# So we disable printing on this platform for now
printing = BSD
```

- On AIX computers, the following comments are added:

```
#
# On AIX, the service NMBD may fail to start because Samba
# cannot determine the correct IP subnet mask.
# In this case, you can manually specify the correct subnet mask.
# For example if you have the following configuration:
#
# Interface     = eth0
# IP Address    = 192.168.97.199
# Subnet mask   = 255.255.252.0
#
# then set the interfaces keyword as follows:
#
# interfaces = eth0 192.168.97.199/255.255.252.0
#
```

Testing Changes to the smb.conf File

When you make changes to the smb.conf file, you should run the Samba utility `testparm` to make sure there are no errors in your smb.conf file before putting it into production use. When you run the `testparm` utility, you should see output similar to the following:

```
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[samba-test]"
Loaded services file OK.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions

[global]
 workgroup = ARCADE
 realm = ARCADE.NET
   security = ADS
 auth methods = guest, sam, winbind, ntdomain
 passdb backend = tdbsam:/etc/samba/private/passdb.tdb
 syslog = 0
 enable core files = No
 server signing = auto
 machine password timeout = 0
 adbindproxy backend = cdc:/usr/share/centrifydc/lib/libcapi.so
 adbindproxy standard mappers = No
 template shell = /bin/bash
 winbind use default domain = Yes

[homes]
 comment = Home Directories
 read only = No
 browseable = No

[printers]
 comment = All Printers
 path = /usr/spool/samba
 printable = Yes
```

```
 browseable = No

[samba-test]
 path = /samba-test
 read only = No
 guest ok = Yes
```

Using adbindproxy.pl

This section describes the options available for the adbindproxy command-line tool. The adbindproxy.pl utility is used to configure Samba and Authentication Service to work together and provides specific functions, such as exporting UIDs and GIDs, creating symbolic links to Samba binaries and libraries, and restoring backed-up Samba files.

> **Note**: For step-by-step instructions about running adbindproxy.pl to configure Samba and the Authentication Service to work together, see Running the adbindproxy.pl Script.

Synopsis

```
adbindproxy.pl [--help] [--info] [--restore] [--unconfig] [--adjoinExtraOptions] [--adleaveExtraOptions] [--version] [--verbose]
adbindproxy.pl [--export] [--groupFile filename] [--userFile filename] [--tdbfile filename]
adbindproxy.pl [--record] [--responseFile filename]
adbindproxy.pl [--nonInteractive] [--responseFile filename]
adbindproxy.pl [--service start|stop|restart|status]
```

adbindroxy.pl Options

You can use the following options with this command:

-c --test filename	Generate a test target Samba configuration file. With this option, the script generates a target Samba configuration file with the filename for review. This option is a review option and does not change any configuration or make any changes.
-E, --export	Export user IDs (UIDs) and group IDs (GIDs) that are stored in Samba's winbindd_idmap.tdb file. Use the--groupFile and --userFile options to specify the export files for the GIDs and UIDs. Use the --tdbfile option to specify the .tdb file that contains the GIDs and UIDs. After export, you can use the Access Manager Console to import the users and groups with their existing UID and GID mappings into a zone.
-f, --responseFile filename	The filename specifies the response file for recording with the -x option or for non-interactive mode with the -n option. If you don't specify a filename, the default is /var/centrify/samba/adbindproxy.pl.rsp.
-g, --groupFile filename	Specify the file in which to write the Samba-created Active Directory group to GID mappings. Use this option with the export option. By default, the file is: /etc/group
-h, --help	Display the adbindproxy.pl usage information.
-i, --info	Display Samba interoperability information.
-j, --adjoinExtraOptions adjoinoptions	The adjoinoptions are the additional options to be used for the adjoin command. Do not specify the domain or the following options with adjoinExtraOptions, because they're already handled in the response file: -u / --user -c / --container -V / --verbose -n / --name -s / --server -T / --trust -k / --des -z / --zone -a / --alias
	The adleaveoptions are the additional options to be used for the adleave command.

-l, --adleaveExtraOptions adleaveoptions	Do not specify the domain or the following options with adleaveExtraOptions, because they're already handled in the response file: -u / --user -f / --force
-n, --nonInteractive	Run adbindproxy.pl in non-interactive mode using the response file. It is recommended to have the machine joined to the Active Directory domain before running this script in non-interactive mode. Otherwise, adbindproxy.pl needs to obtain the Active Directory authorized user password from the command line with the -j/-l option, or interactively from the terminal. **WARNING**: Typing the password in the command line NOT secure, do NOT do that unless you know what you are doing.
-r, --restore	Restore files backed up from the first time you configured Samba for interoperability with the Authentication Service. Typically, you run adbindproxy.pl with the restore option to restore Samba files before uninstalling the integration components that were provided in adbindproxy.
-S, --symbol	Force the creation of symbolic links to Server Suite for Samba binaries and libraries without asking for confirmation.
--s, --service <start|stop|restart|status>	Control the CentrifyDC Samba service. If you haven't configured the CentrifyDC Samba service yet, this option has no effect. If you specify --service status, there will be a return value of 0 if the service is running and a return value of 1 if the service isn't running.
-T, --noTestShare	Specify to not create the test folder "/samba-test" and not add the "samba-test" share when updating the smb.conf file.
-t, --tdbFile filename	Specify the location of the winbindd_idmap.tdb file that contains Samba UID and GID information. This option is used during the UID and GID export process. If you omit this option, the default file to export from is: /var/lib/samba/winbindd_idmap.tdb
-u, --userFile filename	Specify the file in which to write Samba-created Active Directory user to UID mappings. Use this option with the -exports option. By default, the file is /etc/passwd.
-v, --version	Display version information for the installed software.
-V, --verbose	Display detailed information for each operation.
-x, --record	Record the user input into the response file which can be used later in non-interactive mode.

Examples

To display basic information about the configuration of the Samba integration and interoperability with authentication service and Active Directory, you could type a command line similar to the following:

```
adbindproxy.pl --info
```

This command displays information similar to the following (where v.v.v is the Server Suite version number and s.s.s is the Samba number):

```
The Samba base path is: /usr
CentrifyDC version = CentrifyDC v.v.v
CentrifyDC Architecture = 64-bit
CentrifyDC Realm = ARCADE.NET
CentrifyDC NTLM Domain = ARCADE
CentrifyDC Host = magnolia.arcade.net
CentrifyDC Short Host = magnolia

Samba Version = s.s.s
Samba Architecture = 64-bit
Samba Realm = ARCADE.NET
Samba NetBIOS Name = MAGNOLIA
```

```
Samba Version Supported = yes
Samba and CDC in same Realm = yes
Samba and CDC share machine account = yes
Password sync using libtdb = <not specified>
```

To export existing Samba GID and UID information that you want to import into a Server Suite Zone, and to show details about the operation performed, type a command line similar to the following:

```
adbindproxy.pl --export --verbose
```

This command displays information similar to the following:

```
The existing UID mappings have been exported to
/var/centrify/samba/passwd.

The existing GID mappings have been exported to
/var/centrify/samba/group.
```

To record the user input to a response file:

```
# adbindproxy.pl -x
```

To run adbindproxy.pl in non-interactive mode with the response file that was generated previously at the default location:

```
# adbindproxy.pl -n
```

Developer Tools

- Adedit Command Reference and Scripting Guide
- PowerShell Scripting
- Windows API

This guide describes ADEdit for UNIX administrators who want to manage Server Suite and Active Directory from a Linux, UNIX, or Mac computer through CLI commands or scripts. It assumes that you are well-versed in Active Directory's architecture and management, and that you're equally well-versed in Server Suite access control and privilege management features. For more complete information about Server Suite software and management tasks, see the *Administrator's Guide for Linux and UNIX*.

This *ADEdit Command Reference and Scripting Guide* describes how to use the DelineaADEdit command-line interface to manage Delineaobjects stored in Microsoft Active Directory. ADEdit is a Tool command language (Tcl) application that enables administrators to run commands and write scripts that modify data in Active Directory directly from their Linux or UNIX console.

Intended Audience

This guide describes ADEdit for UNIX administrators who want to manage Delineaand Active Directory from a Linux, UNIX, or Mac computer through CLI commands or scripts. It assumes that you are well-versed in Active Directory's architecture and management, and that you're equally well-versed in Delineaaccess control and privilege management features. For more complete information about Delineasoftware and management tasks, see the *Administrator's Guide for Linux and UNIX*.

Using this Guide

This guide describes how to use ADEdit and provide reference information for all ADEdit commands and the ADEdit library. It does not describe how to write Tcl scripts using ADEdit commands. For a comprehensive explanation of Tcl and its use, see *Tcl and the Tk Toolkit* by John K. Ousterhout and Ken Jones (published by Addison-Wesley).

The chapters provide the following information:

- Getting Started with ADEdit describes the basics of ADEdit command syntax and the logical flow of commands that you need to be familiar with before you begin executing interactive ADEdit sessions or writing ADEdit scripts.
- ADEdit Commands Organized by Type assembles the ADEdit commands into logical groups, corresponding to their usage, and summarizes each command.
- Using the Demonstration Scripts provides script samples for a series of common tasks that you can incorporate into your scripts.
- ADEdit Command Reference provides full command descriptions in alphabetical order.
- ADEdit Tcl Procedure Library Reference describes the Tcl procedures available in the ade_lib Tcl library that use ADEdit commands to perform common administrative tasks.
- Timebox Value Format describes the format of the timebox value used to set hours of the week when a role is enabled and disabled.
- Using ADEdit with Classic Zones summarizes the differences between working with classic and hierarchical zone and lists the commands that are specifically for managing authorization in classic zones.
- Quick Reference for Commands and Library Procedures provides a summary of all ADEdit commands and procedures, including the command syntax and abbreviations.

Viewing Command Help

ADEdit provides brief help text for each command. To view the help, enter help command_name from the ADEdit command prompt. For example, to see the help for the validate_license command you would enter the following:

```
>help validate_license
```

You can also display the general help text for ADEdit by entering man adedit from the shell.

Introduction

Centrify ADEdit is a command-line interface (CLI) utility that enables UNIX administrators to manage Centrify objects—such as zones, rights, and roles—in Microsoft Active Directory. This chapter introduces you to ADEdit's main features and architecture.

How ADEdit uses Tcl

ADEdit is implemented as a Tcl application. Tcl (Tool Command Language) is a powerful but easy to learn programming language that provides full scripting ability. With Tcl, administrators can write simple management scripts that perform complex tasks with a single execution. Experienced Tcl programmers can also include ADEdit commands in their own Tcl applications to add Centrify management capabilities and GUI interfaces for ADEdit operations to those applications.

Administrators who aren't familiar with Tcl can use ADEdit as a scripting tool on their Linux or UNIX computer to manage Centrify directly from the command line or by combining commands into scripts.

What ADEdit Provides

The purpose of ADEdit is to let an administrator with the proper Active Directory permissions fully manage Centrify objects from a UNIX console. By using ADEdit, for example, an administrator working on a Linux computer can perform common administrative tasks such as create a new user account, add a user to a new group, or assign a user to a new role. That same administrator might also query Active Directory for information about zones, groups, roles, or any other Centrify objects.

Because ADEdit is a more powerful and flexible tool, it is intended to replace some of Centrify's previous-generation UNIX command line programs such as adupdate and adquery. Those previous-generation tools limited the operations administrators could perform to a computer's currently joined zone and domain. With ADEdit, administrators can manage objects in any zone or domain and perform operations on many more features than were possible using its predecessors.

To give administrators additional flexibility for performing administrative tasks, ADEdit also allows for multiple modes of execution and provides its own accompanying library of predefined scripts for common tasks.

Administration Across Domains and Forests

ADEdit offers complete control of Centrify objects and properties from a Linux or UNIX console. Administrators with the proper permissions on the Active Directory domain controller can modify every aspect of operation that the Access Manager offers. For example, administrators can use ADEdit to create zones, add groups, delegate permissions, define roles, and modify user properties, group membership and role assignments.

ADEdit can operate on any domain in any forest. Its host computer does not need to be joined to a domain to work with that domain. As long as the administrator has the necessary authentication and rights to work on a domain, ADEdit can bind to the domain and work on it. ADEdit can also work simultaneously on multiple domains in multiple forests.

ADEdit enables you to manage all aspects of the access control and privilege management features of multiple Centrify software from a single CLI tool. For example, it can replace adupdate and adquery and offers the features of LDAP clients such as ldapsearch, without the limitations of those command line programs.

Options for Execution

ADEdit offers multiple modes of execution:

- **Interactive mode**. In interactive mode, ADEdit executes single CLI commands in real time. You can enter a series of commands within a shell to perform simple administrative tasks. ADEdit offers command history that is persistent from session to session. You can use the up arrow and Enter keys to review and re-enter commands instead of retyping complete commands from scratch.
- **Script execution**. ADEdit can accept and execute a Tcl script file that includes ADEdit commands. The Tcl scripting language includes full programming logic with variables, logical operators, branching, functions (called procedures in Tcl), and other useful program-flow features. As the script executes, ADEdit keeps the Active Directory objects that it is working on in internal memory. It does not require repeated queries to Active Directory as it works on an object.
- **Executable file**. You can set up any ADEdit Tcl script as an executable file that can run by itself on a UNIX platform.

Scripting makes ADEdit a very flexible administration tool. You can use a single script to handle hundreds or thousands of repetitive tasks that would take a very long time to perform through the console. And you can write a set of scripts to quickly and easily check on and respond to current conditions. A script could, for example, create a new zone, read etc/passwd files on UNIX computers in that zone, and migrate all existing UNIX users it finds there into new zone

user accounts. Another script could find users in specified groups and then assign a new role to all users in those groups.

With that power comes responsibility. It's quite possible for an ADEdit script—or even a single ADEdit command—to completely erase Active Directory's contents if used incorrectly. There are, for the most part, no warnings and there is no undo feature if this happens. Only knowledgeable users should use ADEdit, and it is important to test scripts in sample environments before deploying them to the enterprise.

Library of Predefined Procedures

ADEdit installs with an accompanying library of utility procedures called the ade_lib Tcl library. These procedures use ADEdit commands to perform standard administrative operations such as adding zone users to a zone group or creating a new Active Directory user. The procedures in the library also provide examples of how to use ADEdit commands efficiently in Tcl scripts. From these examples, administrators can learn how to use and adapt ADEdit commands in their own custom scripts.

How ADEdit Works with Other Centrify Components

ADEdit is part Centrify Server Suite and works with specific Windows and UNIX components of the Centrify architecture. As described in the Administrator's Guide for Linux and UNIX, Centrify uses Active Directory, which runs in a Windows network, to stores Centrify-specific data such as zone information. To make computers part of an Active Directory domain, administrators deploy a platform-specific Server Suite Agent. After the agent is deployed and the computer joins an Active Directory domain, the computer is a Centrify-managed computer and ADEdit can define, retrieve, modify, and delete Active Directory and Centrify information for that computer.

Active Directory and ADEdit

Active Directory uses multi-master data storage. It replicates directory data on multiple domain controllers throughout a domain. Changes in data on one domain controller are replicated to the other domain controllers in the domain.

To perform virtually any operation, ADEdit must bind to one or more Active Directory domain controllers. ADEdit can then query Active Directory for data within bound domains, retrieve Active Directory objects, modify retrieved objects, create new objects, and delete existing objects. Those objects include all Centrify-specific objects such as zone objects, zone user objects, role objects, and more.

> **Note**: ADEdit is not limited in scope to Centrify-specific information. An administrator with full privileges could define, retrieve, modify, and delete information for any object or attribute in Active Directory.

Managed Computers and ADEdit

For computers to be managed by Centrify, they must have the Server Suite Agent installed and must be joined to an Active-Directory domain. The Server Suite Agent includes the following components that work directly with ADEdit:

- **adclient** is a Centrify process running on a managed computer. The adclient process communicates with Active Directory to make its host computer part of the Active Directory domain. Applications that require authentication and authorization or other services then use adclient to query Active Directory for that information. In most cases, ADEdit connects directly to Active Directory without using adclient. However, there are some commands that use adclient to get information more efficiently than from Active Directory directly.
- **Centrify command line programs** are commands administrators can run on managed computers to control adclient operations and work with the Centrify data stored in Active Directory. ADEdit replaces some of these commands, but occasionally works in conjunction with other commands such as adflush, especially when executing ADEdit commands that work through adclient. For more information about using command line programs, see the Administrator's Guide for Linux and UNIX.

Other Administrative Options

ADEdit is intended to the primary tool for administrators who want to perform administrative tasks directly from a command line or in scripts on Linux, UNIX, and Mac OS X computers. However, there are two other administrative options for performing the same tasks outside of ADEdit:

- The **Access Manager** console runs on a Windows computer and provides a graphical user interface that you can use for complete control of Centrify-related information and some Active Directory features.
- The **Centrify Server Suite SDK for Windows** provides application programming interfaces (API) that you can use to control all of the same features provided the **Access Manager** console.

It's important to realize when using any of these tools that an instance of one of these tools has no knowledge of other tool instances and acts as if it's the only administrative tool at work. For example, if one administrator uses the Access Manager console to modify a zone object at the same time as another administrator uses ADEdit to modify the same zone object, their changes might clash. For example, if the changes are first saved by the administrative using

Access Manager, those change might be overridden by changes saved by ADEdit. The last tool to save object data has the final say.

This is true as well for different instances of ADEdit. If two administrators both use different ADEdit instances simultaneously to work on the same object, the administrator who last saves the object is the only one whose work will have an effect on the object.

It's important when using ADEdit in an environment with multiple administrators to retrieve an object, make changes, and check it back in efficiently to avoid conflicts. ADEdit object changes are not atomic.

It helps to bind all administration tools to the same domain controller within a domain to further minimize conflicts. If tools work on different domain controllers, one tool's changes may take time to replicate to the other domain controllers, so other tools connected to other domain controllers won't be able to see those changes immediately.

ADEdit Components

ADEdit has two components: the ADEdit application and the ade_lib Tcl library. They are both installed when the Server Suite Agent is installed on a Linux, UNIX, or Mac OS X computer to be managed.

A user can access ADEdit through a CLI in a shell or through an executing Tcl script or Tcl application. ADEdit's Tcl interpreter executes the commands it receives from the CLI using the ADEdit commands and Tcl commands that are part of ADEdit. It may also use ade_lib Tcl library commands if specified. Tcl scripts and applications use ADEdit's commands and ade_lib Tcl library commands directly. ADEdit binds to an Active Directory domain controller, with which it exchanges data. ADEdit may also (in a few cases) get data from Active Directory through the adclient process.

The ADEdit Application

ADEdit uses Tcl as its scripting language. Tcl is a long-established extensible scripting language that offers standard programming features and an extension named Tk that creates GUIs simply and quickly. Tcl is described in the authoritative book Tcl and the Tk Toolkit by John K. Ousterhout and Ken Jones (Addison-Wesley, 2010).

ADEdit includes a Tcl interpreter and the Tcl core commands, which allow it to execute standard Tcl scripts. ADEdit also includes a set of its own commands designed to manage Centrify and Active Directory information.

ADEdit will execute individual commands in a CLI (in interactive mode) or sets of commands as an ADEdit script.

The ade_lib Tcl Library

The ade_lib Tcl library is a collection of Tcl procedures that provide helper functions for common Centrify-related management tasks such as listing zone information for a domain or creating an Active Directory user. You can include ade_lib in other ADEdit scripts to use its commands.

To use ade_lib in a Tcl script or in an ADEdit session, begin the script or session with: `package require ade_lib`

ADEdit Context

When ADEdit commands work on Active Directory objects, they don't specify a domain and the object to work on as part of each command. ADEdit instead maintains a context in memory that defines what commands work on.

ADEdit's context has two types of components:

- **A set of one or more bindings that connect ADEdit to domains in the forest**. Each binding uses an authentication to connect to an Active Directory domain controller. The authentication must have enough rights to perform ADEdit's administrative actions on the domain controller. Each binding binds ADEdit to a single domain; multiple bindings bind ADEdit to multiple domains at one time.
- **A set of zero, one, or more selected Active Directory objects that ADEdit works on**. A selected object is typically a Centrify object such as a zone, zone user, role, or NIS map, but can also be any generic Active Directory object. ADEdit stores each selected object with all of its attributes (called fields within ADEdit). ADEdit stores no more than one type of each selected object: one zone object, for example, one PAM application object, one generic Active Directory object, and so on.

An ADEdit session or script typically starts by binding to one or more domains. If ADEdit isn't bound to a domain, none of its commands that work with Active Directory (which is most of them) have any effect. Once bound, ADEdit commands work within the scope of all currently bound domains.

An ADEdit session or script then typically selects an object to work on: it specifies an object such as a zone user object that ADEdit retrieves from Active Directory and stores in memory as part of the context. All subsequent zone user commands then work on the zone user object in memory, not the zone user object as it is stored in Active Directory.

When finished with a selected object, the session or script can simply ignore the object (if nothing has changed in it) or it can save the object back to Active Directory (if the object has been modified and modifications need to go back to Active Directory, overwriting the object there). The selected object remains stored in ADEdit's context until the session or script selects a new object of the same type, which replaces the previous object.

By maintaining a context with selected objects, ADEdit avoids constant Active Directory queries for successive object management commands: A selection command queries Active Directory to retrieve an object. Reading or modifying object fields occurs internally and doesn't require Active Directory queries. If the object is saved, a final Active Directory query returns the modified object to Active Directory.

Context Persistence

ADEdit's context persists for the duration of an ADEdit interactive session. The context in an ADEdit script persists only until the end of the script's execution.

Pushing and Popping Contexts

ADEdit can save and retrieve contexts using push and pop commands that use a stack to store successive levels of context. Pushing and popping contexts is useful within Tcl scripts when jumping to a procedure. The script can push the current context to the stack, create an entirely new context for the procedure, then pop the original context back when exiting the procedure.

Context Cautions

Working with ADEdit's context requires some thought. Commands that affect objects don't explicitly specify an object, so you must be careful to ensure that the correct object is specified before executing commands that affect the object. ADEdit has context reporting commands that help by showing current domain bindings and selected objects.

It's important to realize that any modifications to a selected object have no effect until the object is saved back to Active Directory. If you forget to save an object, you lose all modifications.

If you keep an object in context a long time between selecting the object and saving the object, be aware—as noted earlier—that another administration tool may alter the object in Active Directory during that time and you won't know about those alterations.

Logical Organization for ADEdit Commands

The commands you can execute with ADEdit fall into the following logical categories:

- **General-purpose commands** that control ADEdit operation and provide information about ADEdit. For example, you use these commands to view usage help, set the LDAP query time-out interval, and quit ADEdit.
- **Context commands** that set up and control the ADEdit domain context. For example, you use these commands to bind to a domain before subsequent object management commands, view current bindings, and change the context.
- **Object management commands** that enable you to perform all of the same tasks as you can with Active Directory Users and Computers and Access Manager. For example, you use these commands to create, select, and manage zones, users, groups, computers, rights, roles and role Assignments.
- **Utility commands** that perform useful data retrieval and data conversion tasks. For example, you use these commands to convert domain names and security principal names from one format to another.
- **Security descriptor commands** that modify security descriptors and make them readable.

For example, you use these commands to convert security descriptors strings from one format to another.

For more information about the commands each category, see ADEdit commands Organized by Type.

For details about specific commands, see ADEdit Command Reference.

Getting Started with ADEdit

This chapter describes ADEdit's basic syntax, shows the typical logic flow used to handle Server Suite objects, and describes in detail the steps in that logic flow using simple examples.

Starting ADEdit for the First Time

The ADEdit application (adedit) and accompanying library of Tcl procedures (ade_lib) are installed automatically when you install the Server Suite Agent on a UNIX, Linux, or Mac OS X computer. Therefore, both the application and the library are immediately available on any Server Suite-managed computer. You are not required to join the domain before using ADEdit for the first time.

To start a new interactive ADEdit session, type adedit in a standard shell after logging on to your computer. A new angle bracket (>) prompt indicates that you are in an interactive ADEdit session. For example:

```
[myprompt]$ adedit >
```

Anyone can launch ADEdit. However, only users who have sufficient privileges can modify Active Directory objects and Server Suite-specific data.

Basic Command Syntax

ADEdit includes a Tcl interpreter and uses Tcl syntax. However, ADEdit commands have their own syntax within the Tcl syntax. Like other Tcl commands, ADEdit commands are always completely lowercase. ADEdit does not recognize commands with uppercase characters.

Arguments and Options

An ADEdit command works very much like a UNIX command. Depending on the command, you might be required to specify one or more arguments. An argument is typically a variable that follows the command name to provide data that controls the operation to be performed. In some cases, values for the variables are required for a command to execute. In other cases, variables might be optional. The reference information for individual commands indicates whether arguments are required or optional. In most cases, however, arguments must be entered in the order specified for the command.

In addition to arguments, ADEdit commands may or may not have options. Options must precede a command's arguments. Each option is a single word preceded by a hyphen (-) such as -write. Options can also have their own arguments. If an option takes an argument, it must immediately follow the option.

Options are used to control specific operations of ADEdit commands. For example:

```
>bind -gc acme.com administrator #3gEgh^&4
```

In this example, the bind command has an option -gc that specifies a global catalog domain controller. Three arguments follow the option. The first argument is required and specifies the domain to which to bind. The second and third arguments are optional and provide a use name and password to be used for binding.

Command Execution and Results

Like most UNIX commands, ADEdit produces no output or return value if a command executes successfully. Only commands that are defined to return a result produce output when an operation completes successfully. If a command fails, however, ADEdit notifies you of an error in execution and reports the general reason for failure. For example, you might see an error message indicating the wrong number of arguments or a connection problem.

Some commands return results as a Tcl list that other commands in a Tcl script can use. Other commands output results directly to standard output (stdout) where the results are displayed in the shell. You can redirect a command's stdout output to a file or other destination, if desired.

Commands that return Tcl lists start with get followed by an object type (get_zone_users, for example) and return the list of the objects matching the specified object type that are stored in Active Directory. Because other commands can use the Tcl list to act on the returned data, the get commands are especially useful for writing scripts.

Commands that send data to stdout start with list followed by an object type (list_zone_groups, for example) and return the list of the objects matching the specified object type that are stored in Active Directory for the currently selected context. Because the list goes to stdout, the list commands are especially useful for displaying data in interactive sessions as a script executes.

Using Command Abbreviations

Most ADEdit commands have an abbreviation that you can use in place of the full command name. For example, the command list_zone_users has the abbreviation lszu. You can use either the full command name or the abbreviation for any command.

Using the Command History

ADEdit in an interactive session retains a history of previously entered commands. You can visit the command history by pressing the up arrow key to go back in the history and the down arrow key to go forward. Press Enter to run the current command.

ADEdit retains its command history across sessions, so if you quit ADEdit and restart it, you can still visit commands entered in the previous session. The command history has a 50command capacity. Once full, the history drops old commands as you enter new commands.

Using the Help Command

The ADEdit `help` command provides brief information about ADEdit commands. If you enter `help` in ADEdit followed by a command or command abbreviation, `help` returns information about that command, including its syntax.

You can use the wildcard character `*` to specifying any number of variable characters or `?` to specify a single variable character within a command string following the help command. The `help` command returns `help` text for all commands that match the wildcard string. For example, the following command returns help for all commands that start with `get`.

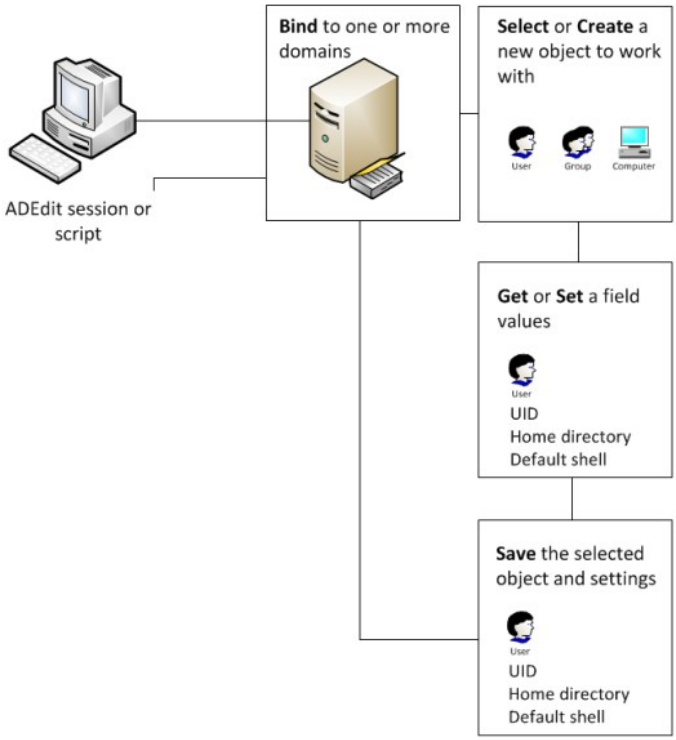
```
> help get*
```

Learning to Use ADEdit

You can use ADEdit interactively to run individual commands or to execute scripts directly. You can use ADEdit commands in scripts that you convert into executable files that can be execute outside of ADEdit sessions. Because scripts can automate and simplify many administrative tasks, it is important for you to know how to combine ADEdit commands in the proper sequence to get the results you are looking for.

Before you begin writing scripts that use ADEdit commands, you should be familiar with the most common logical flow for managing Server Suite-specific and Active Directory objects.

The following illustration provides an overview of the logical process.



As illustrated, the typical logic flow in a ADEdit script follows these steps:

1. **Bind** ADEdit to one or more domains within a forest.

   The domains to which you bind will define the logical boundaries within which all subsequent commands work.

2. **Select** an existing Active Directory object or create a **new** object with which to work.

   You can use select commands to retrieve existing object from Active Directory and store them in memory. You can use new commands to create new objects of a specified type and store them in the ADEdit context as the currently selected object.

   There are also create commands that create a new objects in Active Directory without putting the object in the ADEdit context. You must explicitly select objects that are created with create commands.

3. **Get** or **set** values for a selected object.

   After you select an object to work with and it is stored in memory—that is, the object is in the ADEdit context—you can read field values to see their current settings or write field values to change their current state.

4. **Save** the selected object and any settings you changed.

   If you modify an object in memory or you have created a new object in memory, you must save it back to Active Directory for your changes to have any effect.

As these steps suggest, ADEdit is very context-oriented. The bindings you set and the objects you select determine the ADEdit current context. All commands work within that context. If you select a zone, for example, subsequent commands use the selected zone as the context in which to add new zone users, zone computers, and zone groups.

Outside of scripts that perform the most common administrative tasks, you might use ADEdit commands differently and without following these steps. For example, you might use ADEdit to convert data from one format to another, view help, or get information about the local computer without following the typical logic flow, but those tasks would be exceptions to the general rule.

Binding to a Domain and Domain Controller

ADEdit must bind to one or more domains before any ADEdit commands that affect Active Directory objects will work. When you execute the bind command, you specify the domain to which to bind. You can also specify a user name and password for the bind operation to provide authentication.

The domain can be any domain in the current forest. The ADEdit host computer does not have to be joined to a domain to bind to and work with a domain. A binding command can be as simple as:

```
>bind acme.com
```

If you specify a domain with no options, ADEdit automatically finds the domain's closest, fastest domain controller. Options can narrow down the choice of domain controllers. The -write option, for example, specifies that you want ADEdit to choose a writable domain controller. The -gc option specifies that ADEdit use the global catalog (GC) domain controller. You can use both options to choose a writable GC domain controller, for example:

```
>bind -write -gc acme.com
```

Alternatively, you can name a specific domain controller as a part of the domain name:

```
>bind dcserv1@acme.com
```

> **Note**: Active Directory is a multi-master LDAP system. Changes made at any one domain controller eventually propagate to all other domain controllers in the domain (if they're universal changes). If any administration tools—such as Active Directory Users and Computers, Access Manager, or other instances of ADEdit—bind to the same domain controller, changes made by any one of the tools are immediately available to the other tools without waiting for propagation.

Authentication

If no credentials are provided with a bind command, ADEdit gets its authentication data from the Kerberos credentials cache if one exists. Alternatively, you can provide a user name or both a user name and password. For example:

```
>bind acme.com administrator {e$t86&CG}
```

Notice that the password is enclosed in braces ({}) to ensure that Tcl handles it correctly. Without the braces, Tcl syntax will automatically substitute for some characters such as the $ used in the password. For example, a dollar sign specifies the contents of a variable in Tcl. Enclosing a string in braces guarantees that Tcl will not try to substitute for any of the characters in the string. Tcl drops the braces when it passes the string on.

You can also use the credentials of the ADEdit's host computer by using the -machine option:

```
>bind -machine acme.com
```

> **Note**: Whatever credentials you use, they must be for an account on the Active Directory domain controller with enough authority to read from and make changes to Active Directory objects in the domain. Without the proper authority, ADEdit commands that use Active Directory won't work.

Binding Scope and Persistence

Binding to a single domain allows ADEdit commands to work on Active Directory in that domain. You can bind to multiple domains to allow ADEdit commands to work on more than one domain. To bind to multiple domains, you simply use multiple bind commands, one for each domain.

Once bound to a domain, ADEdit remains bound to that domain until another binding occurs to the same domain (possibly using a different authentication or specifying a different domain controller) or until the current interactive session or executing script ends. Binding might also end if the current context is popped and ADEdit reverts to an earlier context without the binding.

Binding and Join Differences

The ADEdit bind operation is not the same as having the ADEdit host computer join an Active Directory domain. A join is the adclient connection to Active Directory for the host computer. A computer is only allowed to join one domain. A bind is an ADEdit connection to Active Directory, and it can be to more than one domain in the forest. The binding is completely independent of the host computer's joined domain.

> **Note**: A few ADEdit commands that start with joined_* use adclient to retrieve data from Active Directory. Those commands are affected by the host computers's joined domain because they require adclient to be connected to Active Directory and can only get data from the joined domain.

Controlling Binding Operation

You can control the way ADEdit's binding to Active Directory operates. The set_ldap_timeout command sets a time interval for ADEdit's LDAP queries to execute by Active Directory. ADEdit considers a query that doesn't execute by the time-out interval as failed.

Selecting an Object

ADEdit manages Server Suite information by working with the objects in Active Directory. The Server Suite-specific object types are:

- Zones
- Zone users
- Zone computers
- Zone groups
- Roles
- Role assignments
- Privileged UNIX command rights
- PAM application rights
- NIS maps

However, you are not limited to using ADEdit only for managing Server Suite-specific object types. You can also use ADEdit commands to work with generic Active Directory objects, including computers, users, groups, and other classes.

Selection Commands

The ADEdit object select commands have the form select_xxx where xxx is an object type. When you select an object (select_zone, for example), ADEdit looks for the object specified in Active Directory and retrieves it to store the object in the current context.

Each select command is tailored to the type of object it retrieves. As an example, after binding to acme.com, you can use a get_zones command to list the zones in the bound domain, then use a select_zone command to select the zone you want to work with:

```
>get_zones acme.com
{CN=default,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com}
{CN=cz1,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com}
{CN=cz2,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com}
{CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com}
>select_zone {CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com}
```

As this example illustrates, each zone is list by its distinguished name (DN) and you use the distinguished name to identify the zone you want to use.

Selection as Part of Context

Once an object is selected, it resides in memory (context) with all attendant field values. Further ADEdit commands can examine and modify the object in context.

ADEdit keeps only one selected object of each type in context at a time. If you select or create another object of the same type, the new object replaces the old object in memory without saving the old object to Active Directory. ADEdit can and does keep multiple objects in context, but each object must be a different type.

> **Note**: A currently selected object often affects work on other objects types, especially the currently selected zone. For example, if you select a zone user, you must first select a zone so that ADEdit knows in which zone to look for the zone user. If you don't first select a zone, you can't select and work on various zone objects such as zone users, zone computers, and zone groups. Knowing your context as you work on objects is important.

Persistence

A selected object stays selected until another object of the same type replaces it or until the current interactive session ends or executing script ends. When an ADEdit session ends, all selected objects are removed from ADEdit's memory. In most cases, you must explicitly save changes to objects in memory to ensure the changes are stored in Active Directory.

Creating a New Object

You can use ADEdit new_xxx commands, where xxx is the object type, to create new objects to work on instead of selecting existing objects. When you use new_xxx commands, commands, ADEdit creates an object of the specified type and stores the object as the currently selected object of that type in ADEdit's current context.

In most cases, ADEdit does not provide default values for a new object's fields. If you create a new object, its fields are empty. You can use the ADEdit set_xxx commands to set values for the fields that are specific to each object type.

Here are some notes about creating objects in ADEdit:

- Creating a new zone works differently than all other object types: ADEdit does not create a new zone in memory. ADEdit creates new zones directly in Active Directory and fills in zone fields with default values. After you create a zone, you must then select it to examine and modify it.
- ADEdit cannot create AIX extended attributes in a Microsoft Services for UNIX (SFU) zone (Ref: CS-25392c).
- Some non-alphanumeric characters are valid for Windows user or group names and are converted to underscore ("_") when changed to be UNIX names in the Access Manager, but cannot be used in adedit. (Ref: IN-90001) The following characters cannot be used in adedit: \ ( ) + ; " , < > =

Examining Objects and Context

The ADEdit context is a combination of current bindings and currently selected objects. You can examine the properties of currently selected objects using ADEdit get_xxx or list_xxx commands, where xxx is an object type. For example, you can use the get_roles or list_roles command to see a list of roles in the current zone.

Getting Field Values for Objects

You can also use get_xxx_field commands to retrieve field values for different types of objects. For example:

```
>select_zone_user adam.avery@acme.com
>get_zone_user_field uname
adam
```

In this example, ADEdit retrieves the value of the field uname—in this case, the UNIX user name field—for the currently selected zone user adam.avery@acme.com.

Getting Current Context Information

You can examine ADEdit's current context at any time using two different commands: the show command and the get_bind_info command.

The show command returns all bindings and selected objects in the current context. For example:

```
>show
Bindings:
    acme.com: calla.acme.com
```

```
Current zone:
    CN=global,CN=Zones,CN=ACME,CN=Program Data,DC=acme,DC=com
Current nss user:
    adam.avery@acme.com:adam:10001:10001:%{u:samaccountname}:%{home}/%{user}:%{shell}:
```

You can use optional arguments to limit the information the show command returns.

The get_bind_info command returns information about a bound domain. When you use this command, you specify the information you want to retrieve, such as the domain's forest, the name of the current domain controller, the domain's security identifier (SID), the functional level of the domain, or the functional level of the domain's forest. For example:

```
>get_bind_info acme.com server
adserve02.acme.com
```

In this case, ADEdit returns the name of the bound server for the domain acme.com.

Modifying or Deleting Selected Objects

Once an object is selected and residing in the ADEdit context, you can modify its fields using the ADEdit set_xxx_field commands, where xxx is the object type. These commands allow you to specify a field name and a field value. For example:

```
>select_zone_user adam.avery@acme.com
>set_zone_user_field uname buzz
```

This example selects the zone user adam.avery@acme.com and sets the uname field for the zone user—the UNIX user name—to buzz. The field is set to the new value only in memory., however. You must save the object before the new field value is stored in Active Directory and takes effect within the object's domain. For example:

```
>save_zone_user
```

Deleting an Object

You can delete a currently selected object using the ADEdit delete_xxx commands, where xxx is the object type. When you delete an object, it is deleted from both memory *and* Active Directory. For example:

```
>select_zone_user adam.avery@acme.com
>delete_zone_user
```

This example deletes the currently selected zone user, adam.avery@acme.com, from the he ADEdit context so there's no longer a selected zone user. The command also deletes the zone user object associated with the user adam.avery@acme.com so there's no longer a zone user by that name in Active Directory.

> **Note**: There is no undo for a delete command. Once the object is deleted from Active Directory, you must recreate it if you want it back. Be especially careful if you set up an ADEdit script to delete multiple objects.

Saving Selected Objects

Any new or modified object in ADEdit's context has no effect until you save the object back to Active Directory. You do so using a save_xxx command where xxx is the object type. For example:

```
>save_zone
```

This example saves the currently selected zone object back to Active Directory along with any field values that have been modified since the zone was selected.

Saving an object does not deselect the object. It remains the selected object in memory so that you can further read and modify it.

Pushing and Popping Context

There are times when you may want to save ADEdit's current context, change it to a new context to work on different objects in different domains, and then revert back to the original context. This is particularly true when writing Tcl scripts with subroutines, where you may want to feel free to complete a completely new context without altering the context of the calling code.

ADEdit offers a push and a pop command to save and retrieve contexts to a stack maintained in memory. push saves the complete current context—all of its bindings and selected objects—to the stack. Subsequent push commands save more contexts to the top of the stack, pushing the older contexts further down the stack, allowing for nested subroutines.

pop reads the context from the top of the stack and restores it to memory as the current context. pop also removes the restored context from the stack. Subsequent pop commands pop more contexts off the stack until the stack is empty, at which point pop returns an error.

Creating ADEdit Scripts

You can combine ADEdit commands into scripts that perform many common administrative tasks, such as creating new zones, adding users to zones, or pre-creating computer accounts. After you create a script, you can execute it from a shell that calls adedit or convert it to an executable file that can run directly from the command line.

Starting with a Simple Script

If you are new to scripting, Tcl, or both, you might want to experiment first with a few simple commands before trying to develop scripts that perform administrative tasks. The steps in this section are intended to help you get started.

If you are already familiar with scripting languages or with using Tcl, you might want to skip ahead to the discussion of the sample scripts or directly to the command reference.

To write a simple ADEdit script:

1. Open a new file—for example, my_adedit_script—in a text editor.
2. Type the following line to set up the adedit environment and include the ADedit Tcl library:

   ```
   #!/bin/env adedit
   package require ade_lib
   ```

   If your version of Linux or UNIX has the env command in a location other than the /bin directory, modify the first line to specify that directory. For example, another common location for the env command is /usr/bin. In this case, you would type:

   ```
   #!/usr/bin/env adedit
   ```

3. Type an appropriate bind command to identify the Active Directory domain or domains to use.

   ```
   bind pistolas.org maya.garcia {$m1l3s88}
   ```

   Depending on whether you are going to run this script interactively or as an executable file, you might include or exclude authentication information.

4. Type the appropriate commands to create and select a new zone.

   ```
   `create_zone` tree “cn=sample,cn=zones,ou=acme,dc=acme,dc=com” std
   select_zone “cn=sample,cn=zones,ou=acme,dc=acme,dc=com”
   ```

5. Type the command to list the current zones to stdout to verify the new zone.

   ```
   list_zones pistolas.org
   ```

6. Type the command to save the zone and quit.

   ```
   save_zone
   quit
   ```

7. Save the text file and execute it using ADEdit or as an executable file.

   After you have tested the basic script, you edit it to create new zones, make a zone a child zone, add new zone computers, groups, or users. for example, you might add lines similar to these:

   ```
   new_zone_user AD_user_UPN
   set_zone_user_field field value
   save_zone_user
   list_zone_users
   ```

   If your sample script creates and selects a zone successfully, you should delete or rename the zone each time you iterate through the execution.

The following is a sample of what the simple script might look like:

```
#! /bin/env adeditpackage require ade_lib
bind pistolas.org maya.garcia {$m1l3s88}
`create_zone` tree "cn=test6,cn=zones,ou=acme,dc=pistolas,dc=org" std
select_zone "cn=test6,cn=zones,ou=acme,dc=pistolas,dc=org"
`set_zone_field` parent "cn=US-HQ,cn=zones,ou=acme,dc=pistolas,dc=org"
```

```
list_zones pistolas.org
save_zone
new_zone_user tim@pistolas.org
set_zone_user_field uname tim
set_zone_user_field uid 81000
set_zone_user_field gid 81000
set_zone_user_field gecos "Tim Jackson, Accounting"
save_zone_user
list_zone_users
quit
```

Executing an ADEdit Script using ADEdit

You can execute ADEdit script by invoking ADEdit on the command line or by making the script an executable file and invoking the script itself directly from the command line.

To execute an ADEdit script by invoking ADEdit on the command line

1. Open a shell.
2. Type adedit followed by the name of the script

   For example, if the name of the script is my_adedit_scipt and it is the current working directory, type:

   ```
   adedit my_adedit_script
   ```

   If the script isn't in the current working directory, specify the path to the script and any arguments if the script requires any.

Running an ADEdit Script as an Executable from the Command Line

You can run an ADEdit script without invoking ADEdit first by making the script an executable file.

To run an ADEdit script as a UNIX-executable file

1. Verify the script begins with the following lines:

   ```
   #!/bin/env adedit
   package require ade_lib
   ```

   The script reads it as a comment, however UNIX or Linux will use it to find and execute ADEdit and then execute the rest of the script.

2. Use chmod to make the file executable.

   For example, if the name of the script is my_adedit_scipt and it is the current working directory, type:

   ```
   chmod +x my_adedit_script
   ```

3. Make sure the file's directory is listed in your PATH environment variable if you want to be able to execute the file from any directory.

   Alternatively, modify the script to include the full path to adedit. For example:

   ```
   #!/bin/env /usr/bin/adedit
   ```

   Once set up this way, you can simply enter the script's file name in a shell and have the script execute as a command.

   ```
   /my_adedit_script
   ```

Running an ADEdit Script as a Shell Script

You can also run the script as a shell script. In this case, the script file would have the .sh suffix and would contain the following lines at the beginning of the file:

```
#!/bin/sh
# \
exec adedit "$0" ${1+"$@"}
package require ade_lib
```

ADEdit Commands Organized By Type

As discussed in *Logical Organization for ADEdit Commands*, there are different types of ADEdit commands that can be organized into logical categories. This chapter provides a brief introduction to the ADEdit commands in each of those logical categories. For detailed information about individual commands, see ADEdit Command Reference.

General Purpose Commands

You can use the following general purpose commands to control overall ADEdit operation or return general information about ADEdit or its host computer.

help	Returns information about a specified ADEdit command or all ADEdit commands.
get_adinfo	Returns information about the joined domain, the joined zone, or the name the local computer is joined under.
quit	Quits ADEdit.
set_ldap_timeout	Sets the time-out value used by ADEdit's LDAP commands that perform read and write operations on Active Directory through a binding.

Context Commands

You can use the following context commands set the current domain bindings, report on the current bindings and selected object, and save and retrieve the ADEdit context (which includes both bindings and currently selected objects).

bind	Binds to one or more Active Directory domains to define the ADEdit context for subsequent commands.
get_bind_info	Returns information about the domains to which ADEdit is bound.
pop	Restores the context from the top of the ADEdit context stack.
push	Saves the current context to the ADEdit context stack.
show	Displays the current context of ADEdit, including its bound domains and currently selected objects.
validate_license	Determines whether there is a valid license and stores an indicator in the ADEdit context.

Object Management Commands

You can use object management commands to retrieve, modify, create, and delete Active Directory objects of any kind, including Centrify-specific objects such as zones, rights, and roles. The command set for each object type is similar to the command sets for the other object types.

Zone Object Management Commands

You can use the following zone object management commands to create, select, save, and delete zones and manage zone properties.

create_zone	Creates a new zone in Active Directory.
delegate_zone_right	Delegates a zone administrative task to a specified user or group.
delete_zone	Deletes the selected zone from Active Directory and memory.

get_child_zones	Returns a Tcl list of child zones, computer roles, or computer-specific zones associated with the current zone.
get_zone_field	Returns the value for a specified field from the currently selected zone.
get_zone_nss_vars	Returns the NSS substitution variable for the selected zone.
get_zones	Returns a Tcl list of all zones within a specified domain.
save_zone	Saves the selected zone with its current settings to Active Directory.
select_zone	Retrieves a zone from Active Directory and stores it in memory as the currently selected zone.
set_zone_field	Sets the value for a specified field in the currently selected zone.

Zone User Object Management Commands

You can use the following zone user commands to create, select, save, and delete zone user objects and manage user properties in the currently selected zone.

delete_local_user_profile	Deletes a local user (that is not an Active Directory user) that has a profile defined in the current zone.
delete_zone_user	Deletes the zone user from Active Directory and from memory.
get_local_user_profile_field	Returns the value of a profile field for the currently selected local user (that is not an Active Directory user) that has a profile defined in the current zone.
get_local_users_profile	Returns a Tcl list of profiles for local users (that are not Active Directory users) that are defined in the currently selected zone.
get_zone_user_field	Returns the value for a specified field from the currently selected zone user.
get_zone_users	Returns a Tcl list of the Active Directory names of zone users in the current zone.
list_local_users_profile	Returns a list of local users (that are not Active Directory users) that have a profile defined in the current zone.
list_zone_users	Lists all zone users with NSS data for each user in stdout.
new_local_user_profile	Creates an object for a local user (that is not an Active Directory user) in the currently selected zone.
new_zone_user	Creates a new zone user and stores it in memory as the currently selected zone user.
save_local_user_profile	Saves the object for the currently selected local user (that is not an Active Directory user) after you create the local user object or edit profile field values for the local user object.
save_zone_user	Saves the selected zone user with its current settings to Active Directory.
select_local_user_profile	Selects a local user (that is not an Active Directory user) object for viewing or editing.
select_zone_user	Retrieves a zone user from Active Directory and stores it in memory as the selected zone user.
set_local_user_profile_field	Sets the value of a field for the currently selected local user (that is not an Active Directory user) that has a profile defined in the current zone.

set_zone_user_field	Sets the value for a specified field in the currently selected zone user.

Zone Group Object Management Commands

You can use the following zone group commands to create, select, save, and delete zone group objects and manage group properties in the currently selected zone.

delete_local_group_profile	Deletes a local group (that is not an Active Directory group) that has a profile defined in the current zone.
delete_zone_group	Deletes the zone group from Active Directory and from memory.
get_local_group_profile_field	Returns the value of a profile field for the currently selected local group (that is not an Active Directory group) that has a profile defined in the current zone.
get_local_groups_profile	Returns a Tcl list of profiles for local groups (that are not Active Directory groups) that are defined in the currently selected zone.
get_zone_group_field	Returns the value for a specified field from the currently selected zone group.
get_zone_groups	Return a Tcl list of Active Directory names of all zone groups in the current zone.
list_local_groups_profile	Returns a list of local groups (that are not Active Directory groups) that have a profile defined in the current zone.
list_zone_groups	Lists all zone groups with object data for each group in stdout.
new_local_group_profile	Creates an object for a local group (that is not an Active Directory group) in the currently selected zone.
new_zone_group	Creates a new zone group and stores it in memory as the currently selected zone group.
save_local_group_profile	Saves the object for the currently selected local group (that is not an Active Directory group) after you create the local group object or edit profile field values for the local group object.
save_zone_group	Saves the selected zone group with its current settings to Active Directory.
select_local_group_profile	Selects a local group (that is not an Active Directory group) object for viewing or editing.
select_zone_group	Retrieves a zone group from Active Directory and stores it in memory as the selected zone group.
set_local_group_profile_field	Sets the value of a field for the currently selected local group (that is not an Active Directory group) that has a profile defined in the current zone.
set_zone_group_field	Sets the value for a specified field in the currently selected zone group.

Zone Computer Object Management Commands

You can use the following zone computer commands to create, select, save, and delete zone group objects and manage computer properties in the currently selected zone.

delete_zone_computer	Deletes the zone computer from Active Directory and from memory.

get_zone_computer_field	Returns the value for a specified field from the currently selected zone computer.
get_zone_computers	Returns a Tcl list of Active Directory names of all zone computers in the current zone.
list_zone_computers	Lists all zone computers along with object data for each computer in stdout.
new_zone_computer	Creates a new zone computer and stores it in memory as the currently selected zone computer.
save_zone_computer	Saves the selected zone computer with its current settings to Active Directory.
select_zone_computer	Retrieves a zone computer from Active Directory and stores it in memory as the selected zone computer.
set_zone_computer_field	Sets the value for a specified field in the currently selected zone computer.

Computer Role Object Management Commands

You can use the following computer role commands to create, select, save, and delete computer role objects and manage computer role properties in the currently selected zone.

create_computer_role	Creates a new computer role in Active Directory.
delete_zone	Deletes the selected computer role from Active Directory and memory.
get_role_assignments	Returns a Tcl list of user role assignments associated with the selected computer role.
get_zone_field	Retrieves the computer group associated with the computer role.
list_role_assignments	Lists user role assignments associated with the selected computer role.
new_role_assignment	Creates a new role assignment and associates it with the selected computer role.
save_zone	Saves the selected computer role with its current settings to Active Directory.
select_zone	Retrieves a computer role from Active Directory and stores it in memory as the selected zone for subsequent commands.
set_zone_field	Sets the computer group which is associated with the computer role.

Role Object Management Commands

You can use the following role object commands to create, select, save, and delete role objects and manage role properties in the currently selected zone.

add_command_to_role	Adds a privileged command to the currently selected role.
add_pamapp_to_role	Adds a PAM application right to the currently selected role.
delete_role	Deletes the selected role from Active Directory and from memory.
get_role_apps	Returns a Tcl list of the PAM applications associated with the currently selected role.
get_role_commands	Returns a Tcl list of the privileged commands associated with the currently selected role.

get_role_field	Returns the value for a specified field from the currently selected role.
get_roles	Returns a Tcl list of roles in the current zone.
list_role_rights	List all privileged commands and PAM applications associated with the currently selected role in stdout.
list_roles	Lists all roles in the currently selected zone along with object data for each role in stdout.
new_role	Creates a new role and stores it in memory as the currently selected role.
remove_command_from_role	Removes a privileged command from the currently selected role.
remove_pamapp_from_role	Removes a PAM application from the currently selected role.
save_role	Saves the selected role with its current settings to Active Directory.
select_role	Retrieves a role from Active Directory and stores it in memory as the selected role.
set_role_field	Sets the value for a specified field in the currently selected role.

Role Assignment Object Management Commands

You can use the following role assignment object commands to create, select, save, and delete role assignment objects and manage role assignment properties in the currently selected zone.

delete_role_assignment	Deletes the selected role assignment from Active Directory and from memory.
get_role_assignment_field	Returns the value for a specified field from the currently selected role assignment.
get_role_assignments	Returns a Tcl list of role assignments in the current zone.
list_role_assignments	Lists all role assignments along with object data for each role assignment in stdout.
new_role_assignment	Creates a new role assignment and stores it in memory as the currently selected role assignment.
save_role_assignment	Saves the selected role assignment with its current settings to Active Directory.
select_role_assignment	Retrieves a role assignment from Active Directory and stores it in memory as the selected role assignment.
set_role_assignment_field	Sets the value for a specified field in the currently selected role assignment.

PAM Application Object Management Commands

You can use the following PAM application commands to create, select, save, and delete PAM application objects and manage PAM application properties in the currently selected zone.

delete_pam_app	Deletes the selected PAM application from Active Directory and from memory.
get_pam_apps	Returns a Tcl list of PAM applications in the current zone.

get_pam_field	Returns the value for a specified field from the currently selected PAM application.
list_pam_apps	List all PAM applications along with object data for each PAM application in stdout.
new_pam_app	Creates a new PAM application and stores it in memory as the currently selected PAM application.
save_pam_app	Saves the selected PAM application with its current settings to Active Directory.
select_pam_app	Retrieves a PAM application from Active Directory and stores it in memory as the selected PAM application.
set_pam_field	Sets the value for a specified field in the currently selected PAM application.

Command (dz) Object Management Commands

You can use the following privileged authorization commands to create, select, save, and delete privileged UNIX command and manage command properties in the currently selected zone.

delete_dz_command	Deletes the selected command from Active Directory and from memory.
get_dz_commands	Return a Tcl list of commands in the current zone.
get_dzc_field	Returns the value for a specified field from the currently selected command.
list_dz_commands	List all privileged commands along with object data for each command in stdout.
new_dz_command	Creates a new command and stores it in memory as the currently selected command.
save_dz_command	Saves the selected command with its current settings to Active Directory.
select_dz_command	Retrieve a privileged command from Active Directory and stores it in memory as the selected command.
set_dzc_field	Sets the value for a specified field in the currently selected command.

NIS Map Object Management Commands

You can use the following NIS map commands to create, select, save, and delete NIS maps and manage NIS map entries and properties in the currently selected zone.

add_map_entry	Adds an entry to the currently selected NIS map.
add_map_entry_with_comment	Adds an entry with comments to the currently selected NIS map.
delete_map_entry	Removes an entry from the currently selected NIS map.
delete_nis_map	Deletes the selected NIS map from Active Directory and from memory.
get_nis_map	Returns a Tcl list of the entries in the currently selected NIS map.
get_nis_map_field	Returns the value for a specified field from the currently selected NIS map.

get_nis_map_with_comment	Returns a Tcl list of the entries with their comments in the currently selected NIS map.
get_nis_maps	Returns a Tcl list of NIS maps in the current zone.
list_nis_map	Lists the NIS map entries from the currently selected NIS map in stdout.
list_nis_map_with_comment	Lists the NIS map entries and comments from the currently selected NIS map in stdout.
list_nis_maps	List all NIS maps in the currently selected zone in stdout.
new_nis_map	Creates a new NIS map and stores it in memory as the currently selected NIS map.
save_nis_map	Saves the selected NIS map with its current entries to Active Directory.
select_nis_map	Retrieves a NIS map from Active Directory and stores it in memory as the selected NIS map.

Active Directory Object Management Commands

You can use the following Active Directory commands to create, select, save, and delete NIS maps and manage NIS map entries and properties in the currently selected zone.

add_object_value	Adds a value to a multi-valued field attribute of the currently selected Active Directory object.
delete_object	Deletes the selected Active Directory object from Active Directory and from memory.
delete_sub_tree	Deletes an Active Directory object and all of its children.
get_object_field	Returns the value for a specified field from the currently selected Active Directory object.
get_object_field_names	Returns a Tcl list of the field names for each of the fields attributes associated the currently selected Active Directory object.
get_objects	Performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of matching objects.
new_object	Creates a new Active Directory object and stores it in memory as the currently selected Active Directory object.
remove_object_value	Removes a value from a multi-valued field attribute of the currently selected Active Directory object.
save_object	Saves the selected Active Directory object with its current settings to Active Directory.
select_object	Retrieves an object with its attributes from Active Directory and stores it in memory as the selected Active Directory object.
set_object_field	Sets the value for a specified field in the currently selected Active Directory object.

Utility Commands

You can use the following utility commands retrieve and convert data from format to format, manipulate distinguished names, and manage group membership and user passwords.

dn_from_domain	Converts a domain's dotted name to a distinguished name (DN) format.
dn_to_principal	Searches Active Directory for a DN and, if found, returns the corresponding UPN.

domain_from_dn	Converts a domain's distinguished name (DN) to a dotted name format.
get_group_members	Returns a Tcl list of members in a group.
get_parent_dn	Returns the parent of an LDAP path (a distinguished name): it removes the first element of the DN and returns the rest.
get_pwnam	Searches the etc/passwd file for a UNIX user name and, if found, returns a Tcl list of the passwd profile values associated with the user.
get_rdn	Returns the relative DN of an LDAP path: it returns only the first element of the supplied DN.
get_schema_guid	finds a class or attribute in Active Directory and returns its globally unique identifier (GUID)
getent_passwd	Returns a Tcl list of all entries in the local /etc/passwd file.
joined_get_user_membership	Uses adclient to query Active Directory and returns a Tcl list of groups that a user belongs to.
joined_name_to_principal	Uses adclient to search for a UNIX name and return the security principal associated with that UNIX name.
joined_user_in_group	Uses adclient to check Active Directory to see if a user is in a group.
move_object	Moves the selected object to the specified location.
principal_from_sid	Searches Active Directory for an SID and returns the security principal associated with the SID.
principal_to_dn	Searches Active Directory for a user principal name (UPN) and, if found, returns the corresponding DN.
rename_object	Renames the selected object.
set_user_password	Sets an Active Directory user's password.
sid_to_escaped_string	Converts an Active Directory security identifier (SID) to an escaped string.
sid_to_uid	Converts an Active Directory SID to a user ID (UID).

Security Descriptor Commands

You can use the following security descriptor commands modify SDs and make them readable by humans.

add_sd_ace	Adds an access control entry to a security descriptor.
explain_sd	Converts a security description in SDDL format to a human-readable form.
remove_sd_ace	Removes an access control entry (ACE) from a security descriptor.
set_sd_owner	Sets the owner of a security descriptor.

Using the Demonstration Scripts

This chapter describes the ADEdit sample scripts provided in the package. The scripts are listed in the following table. The corresponding source files are in the /usr/share/centrifydc/samples/adedit directory. The source file name is shown in the table and each script header.

You have a couple of different ways to invoke scripts from the command line (see Creating ADEdit scripts under Getting Started with ADEdit). The sample scripts demonstrate two of them.

Reading command line input	These scripts illustrate two different methods for using the Tcl argv, argc, and argv0 variables.	MktDept.sh getopt-example
Create a parent zone	This script illustrates how to create a Delineaparent zone.	CreateParentZone
Create child zones	This script illustrates how to create two child zones in a parent zone.	CreateChildZones
Create privileged commands and roles	These scripts illustrate how to create new privileged commands and new roles that include those commands.	MakeRole ApacheAdminRole
Add and provision UNIX users	This script and input file illustrate how to add users to Active Directory and copy them to the Active Directory UNIX Users group. If you have the Zone Provision Agent configured and running, you can use this script or one similar to it to automatically provision user profiles when users are added to Active Directory.	AddUnixUsers users.txt
Simple tools	These scripts demonstrate how you can list the computers in a zone, extract field attributes from user objects, and list the users in a zone.	computers-report useracc-report user-report GetComputers
Run a script from a script	These scripts illustrate how you can call a script (setenv) from within another script to perform different queries based on the values entered.	setenv GetChildZones GetGroups GetUsers GetZones

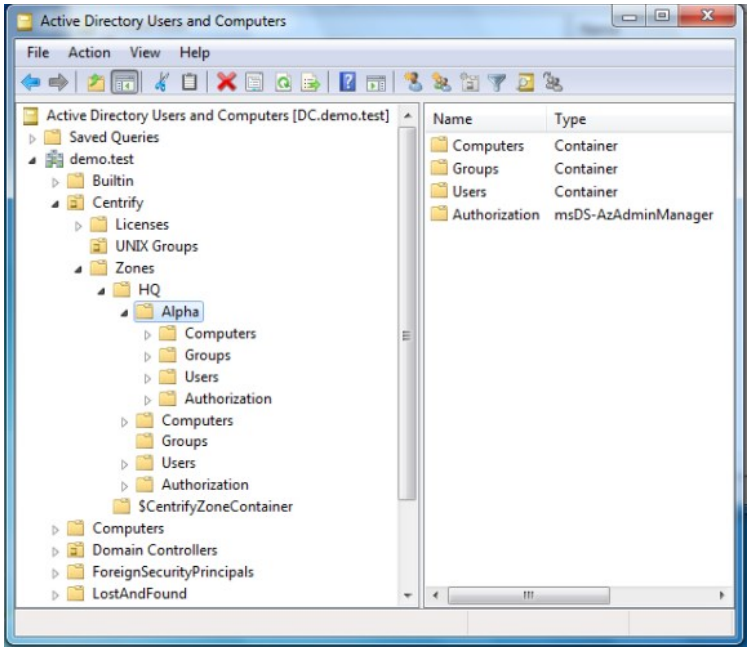
Zone Containers and Nodes

Many ADEdit commands require you to specify the zone container. This container is the root container used by Delineato store the zone information for the users, groups, computers and child zones. This container can have any name and can be anywhere in Active Directory. This container can also be an organizational unit.

Before you proceed, you need to know the location of the zone containers in Active Directory and the distinguished names you use to specify the zone container and its objects.

This section illustrates some sample cases with different locations for the zone container and the distinguished name for commonly used variables in the scripts.

In this example, the installer defined a base organizational unit called Delinea. This architecture is often used because it puts all the UNIX-related information in a single branch. The container with the zone information is called Zones.



In addition to the Zones container location, the installation script requires the installer to specify a location for a container to store the Delinea software licenses. In this figure, the node—Licenses—is in the base organizational unit. However, it does not need to be there.

In this figure, the installer also created another organizational unit called UNIX Groups for the Active Directory groups used for the UNIX users. Keeping all of the groups recreated for the UNIX users in a single node simplifies managing them and the privileges assigned to each user. (With few exceptions, the UNIX users get their rights from the role assigned to the group in which they are a member.) Often, more organizational units are created for managing different classes of UNIX user and UNIX services.

There are two zones in this figure: the parent zone HQ and a child zone named Alpha. Each zone contains nodes labeled Computers, Groups, Users, and Authorization. When you specify a zone, computer, user, or group in an ADEdit command you must use the distinguished name. The following table illustrates the distinguished names.

Domain	demo.test	dc=demo,dc=test
Base organizational unit	Delinea	ou=Acme,dc=demo,dc=test
Zone container	Zones	cn=Zones,ou=Acme,dc=demo,dc=test
Parent zone	HQ	cn=HQ,cn=Zones,ou=Centrity,dc=demo,dc=test
Child zone	Alpha	cn=Alpha,cn=HQ,cn=Zones,ou=Centrity,dc=demo,dc=test
Organizational unit	UNIX Groups	“ou=UNIX Groups,ou=Acme,dc=demo,dc=test”
UNIX group	ApacheAdmins	“cn=ApacheAdmin,ou=UNIX Groups,ou=Acme,dc=demo, dc=test”
Computer in Alpha zone	RHEL	cn=RHEL,cn=Computers,cn=Alpha,cn=HQ,cn=Zones,ou=Centrity,dc=demo,dc=test

You should note that distinguished names can contain space, as illustrated by the UNIX Groups organizational unit. To prevent Tcl from interpreting a space as new element in a list, you can enclose the distinguished name with double quotes (“ “) or using braces (). When specifying distinguished names, you should also be sure to use ou and cn correctly. Commands will fail if you refer to an organizational unit using cn.

Create Tcl Procedures

The following example demonstrates how to create procedures using the Tcl proc command. These two procedures create a new Active Directory user and Active Directory group, respectively, but first check to see if that object already exists in Active Directory.

This example uses the Tcl catch and if commands to determine if the account already exists. catch takes a Tcl script (in this case, the select_object command) and returns a 1 if an error (in this case, the account does NOT exist) occurs. Inside the if command, a non-zero result of the expression causes the body commands (puts and create_aduser or create_adgroup) to be executed. Otherwise, if select_object is successful (the account exists) it does not create the new account.

> **Note**: See the AddUnixUsers script for a similar example that uses the catch and if commands to determine whether a user exists.

Create Active Directory Group Procedure

```
# The following procedure creates an Active Directory group if a
# group with the same distinguished name does not already exist.
proc my_create_adgroup {dn sam gtype} {
        if { [catch {select_object \$dn}] } {
     # If we fail to select the object, the group
     # does not exist. So we create it here.
     puts "Creating \$dn"
     create_adgroup \$dn \$sam \$gtype
  } else {
     puts "\$dn exists. Do not create."
  }
 }
```

Create Active Directory User Procedure

```
# The following procedure creates an Active Directory user if an
# account with the same distinguished name does not already exist.
proc my_create_aduser {dn upn sam pw} {
         if { [catch {select_object \$dn}] } {
     # If we fail to select the object, the account
     # does not exist. So we create it here.
     puts "Creating \$dn"
     create_aduser \$dn \$upn \$sam \$pwd
   } else {
     puts "\$dn exists. Do not create."
   }
 }
```

Reading Command Line Input

In general, Tcl reads the arguments following the script name as a list and creates the following three variables:

- argv: a Tcl list containing all of the arguments in the command line
- argc: a count of the number of arguments in the lists
- argv0: the script name.

For example, the following script uses all three variables. This is a simple command in the form

```
>/bin/sh MktDept.sh name name name
```

where *name* is a person's name, such as Mary or Joe. If you want to use first and last name, surround the name with quotes, for example "Jane Smith".

> **Note**: This code sample demonstrates starting ADEdit from a shell script. The subsequent examples use the executable file model.

MktDept.sh

```
#!/bin/sh
# This script takes a list of names and displays it
#
# \
exec adedit "$0" ${1+"$@"}
package require ade_lib
if { $argc == 0 } {
        puts "Command format: $argv0 name name ..."
        exit 1
}
set total $argc
puts "
The following people are in the marketing department"
while {$total > 0} {
```

```
    incr total -1
    puts "[lindex $argv $total]"
    }
```

The first if statement uses the count, argc, to determine if any arguments have been entered. If the argc value is equal to zero, the user did not enter any names and the script displays the command format message. The argc counter is used again to set the total count of names entered for the while loop. Inside the loop, the names are drawn from the argv list.

Another useful command for parsing command line options is getopt. This command derives from, but is different than, the Tcl getopt command. The ADEdit getopt command has the following syntax:

```
getopt _argv name ?-var?
```

where:

- _argv is the Tcl list that contains the command line arguments.
- name is a label for the associated data.
- ?_var? is the variable name for the data.

For example, the following script illustrates the use of getopt to define the user and group variables that will be used later in the script.

This script also demonstrates how to use a procedure, usage, that prompts the user when she doesn't enter all of the arguments. usage first displays the full command syntax and then the missing argument.

> **Note**: The user and password arguments are optional. If the user enters a user name without the password, the bind program automatically prompts for the password. You do not need to include that prompt in the script.

getopt-example

```
#!/bin/env adedit
# This script takes a domain name and optionally user name and password
# and binds the user to the specified domain.
# If the user does not specify a user name or password, she is prompted.
#
package require ade_lib
proc usage {msg} {
            puts {usage: -d <domain> [-u <user>] [-p <password>]}
             puts $msg
             exit 1
}
if {[getopt argv -d domain] == 0} {
            usage "Missing Domain, ex.   centrify.demo"
}
if {[getopt argv -u user] != 0} {
      if {[getopt argv -p password]} {
            bind $domain $user $password
      } else {
        bind $domain $user}
            } else {
    puts "Enter administrator name:"
            gets stdin user
            bind $domain $user
}
puts "
Binding complete to $domain."
```

Create a Parent Zone

This sample script illustrates how you can create a parent zone. This script uses the puts command to display information and to prompt the user to specify variables that will be used to create the parent zone object. The command line syntax is as follows:

```
>./CreateParentZone - z parentZone -u adminName [-p password]
```

where:

- parentZone is the name of the parent zone you want to create.
- adminName is the name of an Active Directory user with administrator privileges on the domain controller.
- password is the administrator's password. If you do not enter the password in the command line, you are prompted to enter it.

Note that this sample script assumes you are using the default deployment structure with the top-level organizational unit. If you are not using the default deployment structure, you should modify the sample script to reflect the structure you are using before testing its operation.

CreateParentZone

```
#!bin/env adedit
# This script creates a tree zone. Use this, for example, to create the
# parent zone for child zones created in other scripts.
package require ade_lib
proc usage {msg} {
        puts {usage: -z >parentZone> -u >user>}
        puts $msg
        exit 1
}
if {[getopt argv -z parentZone] == 0} {
        usage "Missing the name for the new zone"
}
puts "
Enter the domain name for the bind command"
gets stdin domain
if {[getopt argv -u user] != 0} {
        if {[getopt argv -p password]} {
                bind $domain $user $password
                } else {
                bind $domain $user}
        } else {
        puts "Enter administrator name"
        gets stdin user
        bind $domain $user
}
set domaindn [dn_from_domain $domain]
puts "
Enter the name of the Active Directory container that holds the Delineazone data"
gets stdin zonesNode
puts "
Enter the organizational unit with the Delineazone data container"
gets stdin baseOU
puts "Summary:"
puts "Domain is $domain. DN for the domain is $domaindn"
puts "The base OU is $baseOU."
puts "The container for the zone information is $zonesNode"
puts "The new zone is named $parentZone"
#create the parent zone in Active Directory
puts "
Creating Delineazone $parentZone"
create_zone tree "cn=$parentZone,cn=$zonesNode,ou=$baseOU,$domaindn" std
puts "Created new zone: cn=$parentZone,cn=$zonesNode,ou=$baseOU,$domaindn"
```

Create Child Zones

This script creates two child zones in the domain and parent zone specified in the command line. The command line syntax is as follows:

```
>./CreateChildZones -d domain -z parentZone [-u adminName] [-p password]
```

where:

- domain is the domain name
- parentZone is the name of an existing zone
- adminName is the name of an Active Directory user with administrator privileges on the domain controller
- password is the administrator's password. If you do not enter the password in the command line, your are prompted for it

The *password is optional. If you do not type it in the command line, the script prompts you to enter it.

The script binds to the domain you specify using the user name and password you provide. The script then prompts you to enter the name of the organizational unit and container in which you store the zone information. After that, it prompts you to enter names for the two child zones. Note that this sample script assumes you are using the default deployment structure with the top-level organizational unit. If you are not using the default deployment structure, you should modify the sample script to reflect the structure you are using before testing its operation.

To confirm the script ran successfully, open Access Manager and expand the **Child Zones** node under the parent zone you specified in the command line. If the two new child zones are listed, you can right-click each zone name to see its zone properties.

CreateChildZones

```
#!/bin/env adedit
# This script creates 2 child zones in the domain and parent zone
# specified in the command line.
#
package require ade_lib
```

```
proc usage {msg} {
        puts {usage: -d <domain> -z <parentZone> [-u <user>] [-p <password>]}
puts $msg
         exit 1
}
if {[getopt argv -d domain] == 0} {
        usage "Missing Domain, ex. demo.test"
}
if {[getopt argv -z parentZone] == 0} {
        usage "Missing parent zone, ex. HQ"
}
if {[getopt argv -u user] != 0} {
        if {[getopt argv -p password]} {
                bind $domain $user $password
                 } else {
        bind $domain $user}
        } else {
puts "Enter administrator name"
        gets stdin user
        bind $domain $user
}
puts "
Enter the name of the container for the Delineazone data"
gets stdin zoneContainer
puts "
Enter the organizational unit for the Delineazone data"
gets stdin zoneContainerOU
# Define distinguished name for domain
set domaindn [dn_from_domain $domain]
puts "
Summary:"
puts  "Domain is $domain. DN for the domain is $domaindn"
puts  "The base OU is $zoneContainerOU."
puts  "The container for the zone information is $zoneContainer
"
# Create child zones
puts "Enter child zone name"
gets stdin czone1
puts "
Enter another child zone name"
gets stdin czone2
create_zone tree "cn=$czone1,cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn" std
create_zone tree "cn=$czone2,cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn" std
# link the children to parent
select_zone "cn=$czone1,cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn"
set_zone_field parent "cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn"
save_zone
select_zone  "cn=$czone2,cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn"
set_zone_field parent "cn=$parentZone,cn=$zoneContainer,ou=$zoneContainerOU,$domaindn"
save_zone
puts "
Child zones $czone1 and $czone2 created in $parentZone"
```

Create Privileged Commands and Roles

Users get the rights necessary to run privileged commands and access applications from their role assignments. The predefined UNIX Login role gives users basic access to UNIX computers without any elevated privileges. The scripts in this section illustrate how you can create roles with additional rights. The first sample script uses a separate text file to define a new role and the commands users in that role are allowed to execute. The second sample script illustrates how to define the commands and the role within the script after prompting for bind credentials and the target zone.

Both scripts create the same commands and role.

Privileges and Role Defined in a File

For the first sample script, a single role and its privileged commands are defined in the file Role_apacheAdmin.txt. This sample text file defines the role name and a few sample commands that you might assign to an Apache server administrator. For example:

```
ApacheAdminRole
vi /etc/httpd/conf/httpd.conf
apachectl *
htpasswd *
```

The first line in the Role_apacheAdmin.txt file specifies the new role name. The subsequent lines specify the commands to add to the role. You can edit the text file to suit your environment. For example, you might want add or remove commands or modify the path to the Apache configuration file. To create the role and commands, you can then run the MakeRole sample script and specify the Role_apacheAdmin.txt file name as a commandline argument. The MakeRole sample script then prompts you to enter the domain name, account, and password for the bind command and to type the name of the parent zone where the sample role will be created.

Note that you must specify a parent zone for this sample script. The second sample ApacheAdminRole script, shown in the **Privileges and Roles Defined in the Script** section below, displays the list of zones in the domain to illustrate how you can create a role in a child zone. In addition, this sample script assumes you are using the default deployment structure with the top-level organizational unit. If you are not using the default deployment structure, you should modify the sample script to reflect the structure you are using before testing its operation.

MakeRole

The MakeRole sample script creates a role with the set of privileged commands defined in the sample Role_apacheAdmin.txt file.

```
#!/bin/env adedit
# This script creates a role consisting of a
# set of privileged commands
# The role name and commands are specified
# in a separate file.
#
# The first line in the input file should be
# the new role name.
# The subsequent lines are the names of the
# privileged commands to
# add to the role.
# For example:
#     audit_admin_cmds
#     /usr/bin/vi /etc/security/audit/config
#     /usr/bin/vi /etc/security/audit/objects
package require ade_lib
if { $argc != 1 } {
    puts "usage: $argv0 file"
    exit 1
}
if {[catch {set fp [open [lindex $argv 0] r]} errmsg]}
    {
    puts "Cannot open [lindex $argv 0]."
    exit 1
}
# Get domain and bind
puts "Enter domain name"
gets stdin domain
set domaindn [dn_from_domain $domain]
puts "Enter account name with administrator privileges"
gets stdin administrator
puts "Enter $administrator password"
gets stdin APWD
bind $domain $administrator "$APWD"
# Select the target zone and base organizational unit
puts "Enter the target zone name for the new role"
gets stdin zonename
puts "
Enter the name of the Active Directory container that holds the Delineazone data"
gets stdin zonesNode
puts "
Enter the organizational unit with the Delineazone data container"
gets stdin baseOU
select_zone "cn=$zonename,cn=$zonesNode,ou=$baseOU,$domaindn"
if {[gets $fp line] == -1} {
            puts "Cannot read [lindex $argv 0]."
             exit 1
}
# Create role
puts "Creating role...$line"
set role $line
new_role "$role"
save_role "$role"
set count 0
while {[gets $fp line] >= 0} {
   incr count
# Create command. Each command will be named
# based on the role defined in the first line
# and the command's line number in the file
   set cmd_name $role$count
   new_dz_command "$cmd_name"
   # set the command fields
   set cmd_path $line
   set_dzc_field cmd "$cmd_path"
   set_dzc_field dzdo_runas root
   set_dzc_field umask 077
   # prevent nested execution
   set_dzc_field flags 1
   # save the command
   save_dz_command
   # Add the command to the Role
   add_command_to_role "$cmd_name"
```

```
 }
close $fp
save_role "$role"
```

Privileges and Roles Defined in the Script

In this sample script, you create the same Apache administrator commands and role as the previous script. However, this script displays a list of the zones in the domain and lets you select in which zone to create the commands and role.

ApacheAdminRole

```
#!/bin/env adedit
puts "This script creates privileged commands and the ApacheAdminRole in the zone entered"
package require ade_lib
puts "
Enter the domain name"
gets stdin domain
puts "
Enter the account name to use to modify Active Directory"
gets stdin acctName
bind $domain $acctName
set domaindn [dn_from_domain $domain]
set zonelist [get_zones $domain]
set numberZones [llength $zonelist]
set row 1
set zonenum 0
puts "
This domain contains the following zones"
while {$numberZones != 0} {
        puts "$row. [lindex $zonelist $zonenum]"
        incr zonenum
        incr row
        incr numberZones -1
}
puts "
Enter the row number of the target zone"
gets stdin rowSelect
set zone [lindex $zonelist [incr rowSelect -1]]
select_zone "$zone"
puts "
Creating command-level Apache admin rights in $zone"
puts "
Creating web_edit_httpd_config"
new_dz_command web_edit_httpd_config
set_dzc_field cmd "vi /etc/httpd/conf/httpd.conf"
set_dzc_field description "edit httpd config file"
set_dzc_field dzdo_runas root
set_dzc_field dzsh_runas root
set_dzc_field path /usr/local/apache2/bin
set_dzc_field flags 1
save_dz_command
puts "
Creating web_apachectl"
new_dz_command web_apachectl
set_dzc_field cmd "apachectl *"
set_dzc_field description "Web Apache Server Control"
set_dzc_field dzdo_runas root
set_dzc_field dzsh_runas root
set_dzc_field path /usr/local/apache2/bin
save_dz_command
puts "
Creating web_htpasswd"
new_dz_command web_htpasswd
set_dzc_field cmd "htpasswd *"
set_dzc_field description "Web Apache Manage user files"
set_dzc_field dzdo_runas root
set_dzc_field dzsh_runas root
set_dzc_field path /usr/local/apache2/bin
save_dz_command
#-------------------------------------------------------------------------
# Create ApachedAdminRights role
# The new_role command creates the role in the currently selected zone.
puts "
Creating the ApacheAdminRole with these rights"
# In each role you need to set the sysrights with the set_role_field
# to the following binary values
# password_login = 01
# sso = 02
# ignore_disabled = 04
# full_shell = 08
new_role ApacheAdminRights
add_command_to_role web_edit_httpd_config
```

```
add_command_to_role web_apachectl
add_command_to_role web_htpasswd
set_role_field sysrights [expr 0x0000000b] #full_shell | sso | password_login
save_role
save_zone
```

Add and Provision UNIX Users

It is difficult to provision a lot of UNIX users and ensure that the UID is unique in the domain. To assist you with the process, Delineaprovides a set of features called the Zone Provisioning Agent. The Zone Provisioning Agent includes a service that automatically assigns a unique UID and other UNIX profile attributes, such as the home directory, default shell, and primary GID, based on rules you define.

This script demonstrates how you could use the Zone Provisioning Agent to add and provision users. For this sample script, the list of UNIX users is defined in the source file named users.txt and the Active Directory source group is Unix Users.

> **Note**:To learn more about the Zone Provisioning Agent and automated provisioning, see the *Planning and Deployment Guide*.

users.txt

You specify the names to be added in a text file in which each name is on a separate line. Be sure to use line feed only as the end-of-line; do not use CR-LF. The sample file in the distribution package contains the following names:

```
Amy.Adams
Brenda.Butler
Dennis.Day
Eric.Edwards
```

AddUnixUsers

In the following script, you specify the file name with the user names in the command line. The script then prompts you for the additional information required. The target Active Directory group—Unix Users—is hard-coded into the script.

This script uses the Tcl catch command three times to control processing when an error occurs.

- In the first case, it is used to exit gracefully if the specified file cannot be opened.
- In the second case, catch is used to determine if the user already exists. An error here indicates that the user does not exist and, rather than exiting, the else statement creates the user. (If the user already existed, you would not want to create another Active Directory account.)
- In the third case, catch is used to exit gracefully if the user is already a member of the Unix Users group.

```
#!/bin/env adedit
# This script creates an Active Directory account
# for each user the specified
# and adds the user to UNIX Users group.
# This automatically fills in their UNIX profile.
# Command line input: file name w/ user names in
# format ffff.llll only
# Prompted input: domain, administrator
#name, default password
package require ade_lib
if { $argc != 1 } {
    puts "usage: $argv0 file"
    exit 1
}
if {[catch {set users [open [lindex $argv 0] r]}
      errmsg]} {
    puts "Cannot open [lindex $argv 0]."
    exit 1
}
# Get domain and bind
puts "Enter domain name"
gets stdin domain
set domaindn [dn_from_domain $domain]
puts "Enter account name with administrator privileges"
gets stdin administrator
puts "Enter $administrator password"
gets stdin APWD
bind $domain $administrator "$APWD"
puts "
Define password to be used for all accounts"
gets stdin pwd
# Now start creating accounts from users
# example: "cn=Ellen Edwards,cn=Users,$domaindn"
# "Ellen.Edwards@$domain" ellen.edwards pwd
while {[gets $users sam] >= 0} {
```

```
    set name [split $sam .]
    set dn "cn=[lindex $name 0] [lindex $name 1],
          cn=Users,$domaindn"
    set upn $sam@$domain
    if { [catch { select_object $dn }] } {
        # If we fail to select the object,
        # most probably it
        # does not exist. So we create it here.
        puts "Creating $dn"
        create_aduser $dn $upn $sam $pwd
    } else {
        puts "$dn exists. Skip creating."
    }
# Because we already installed and started ZPA,
# this provisions the
# Active Directory account
    catch { add_user_to_group $sam@$domain
          "UNIX Users@$domain" }
  }
close $users
```

Simple Tools

The following scripts are simple "utilities" for getting information from Active Directory about the managed computers and users accounts:

- computers-report: Lists the managed computers in the zone.
- useracc-report: List the Active Directory users in the domain and several account properties.
- user-report: Lists the users in a zone.
- GetComputers: Lists all of the managed computers in the specified domain and the zone to which each computer is joined.

Following these scripts are sample scripts that demonstrate how you can use a script that calls, for example, commonly-used commands in other scripts. For more information, see **Run a Script from a Script** below.

computers-report

Use this command to list managed computers in the zone. The command line arguments are as follows:

-domain	required	Domain name.
-m	optional	Bind using the ADEdit host computer's credentials (see bind). You can use either the computer credentials (-m) or the user account credentials (-u).
-u	optional	Administrator's account name.
-p	optional	Administrator's account password. **Note**: If you do not enter the password in the command line you will be prompted to enter it.
-sep	optional	Separator used between data. The default is separator is the pipe (|) character.

```
#!/bin/env adedit
# This script lists the managed computers on the zone.
# Command line input is the domain, the
# administrator account name and
# the separator to use between computer's field
# values in the output
package require ade_lib
# Lists all of the managed computers and the zone
proc usage {msg} {
     puts {usage: -domain <domain> [-m] [-u <user>]
       [-p <password>] [-sep csv | tab | <char>]}
     puts $msg
     exit 1
}
if {[getopt argv -domain domain] == 0} {
    usage "Missing domain"
}
set verbose 0
if {[getopt argv -v]} {
```

```
    set verbose 1
}
set sep "|"
getopt argv -sep sep
if {$sep == "csv"} {set sep ","}
if {$sep == "tab"} {set sep "\t"}
if {[getopt argv -m]} {
     bind -gc -machine $domain
} else {
   if {[getopt argv -u user]} {
      if {[getopt argv -p password]} {
          bind -gc $domain $user $password
       } else {
        bind -gc $domain $user}
     } else {
    bind -gc $domain
  }
}
# this code runs entirely off the GC
cache on
set scps [get_objects -gc -depth sub [dn_from_domain $domain] {(&amp;(displayName=$CimsComputerVersion*)(objectClass=serviceConnectionPoint))}]
foreach scp $scps {
    select_object -gc $scp
    set name [get_object_field name]
    set parent ""
    # first look for parentLink
    foreach k [get_object_field keywords] {
       set bits [split $k ':']
       if {[lindex $bits 0] == "parentLink"} {
          set sid [lindex $bits 1]
          #ok we got it
          # make sure it exists
        catch {set parent [principal_from_sid $sid]}
        }
     }
     # if we didn't then try by managed By (DC3)
     if {$parent == ""} {
        set mb [get_object_field managedBy]
        if {$mb != ""} {
           set parent $mb
           }
        }
         set orphan 0
         if {$parent == ""} {set orphan 1}
           set path [get_parent_dn [get_parent_dn
               [get_object_field dn]]]
           set zone [string range [get_rdn $path] 3 end]
           puts $name$sep$zone$sep$orphan
     }
```

useracc-report

Use this command to list all users and their Active Directory account control values. The command line arguments are as follows:

-domain	required	Domain name
-m	optional	Bind using the ADEdit host machine's credentials (see bind) Note: If you use -m you do not need to enter -u
-u	optional	Administrator's account name.
-p	optional	Administrator's account password. Note: If you do not enter the password in the command line you will be prompted to enter it.
-sep	optional	Separator used between data. Default is |

```
#!/bin/env adedit
# This script lists all the users and their Active Directory account control values
package require ade_lib
# List users and the following field
proc usage {msg} {
            puts {usage: -domain <domain> [-m] [-u <user>] [-p <password>] [-sep csv | tab | <char>]}
            puts $msg
            exit 1
```

```
}
if {[getopt argv -domain domain] == 0} {
        usage "Missing domain"
}
set verbose 0
if {[getopt argv -v]} {
        set verbose 1
}
set sep "|"
getopt argv -sep sep
if {$sep == "csv"} {set sep ","}
if {$sep == "tab"} {set sep "\t"}
if {[getopt argv -m]} {
        bind -machine $domain
} else {
   if {[getopt argv -u user]} {
      if {[getopt argv -p password]} {
        bind $domain $user $password
       } else {
        bind $domain $user}
        } else {
    bind $domain
  }
}
cache on
proc my_convert_msdate {msdate} {
    if {$msdate==9223372036854775807} {
       return -1
    }
    return [clock format [expr ($msdate/10000000)-11644473600] -format "%m/%d/%y %H:%M:%S"]
}
proc nice_date {date} {
 if {$date == ""} {return $date}
 if {$date == 0} {return ""}
 set ret [my_convert_msdate $date]
 if {$ret == -1} {return ""}
 return $ret;
}
set users [get_objects -depth sub [dn_from_domain $domain] "(objectcategory=Person)"]
foreach user $users {
        select_object $user
        set uac [get_object_field userAccountControl]
        if {$uac == ""} {continue}
        # gof is get_object_field
        eval "set name [gof cn]"
        #puts [gof dn]
        set sam [gof sAMAccountName]
        set exp [nice_date [gof accountExpires] ]
        set locked [nice_date [gof lockoutTime] ]
        set lastlogon [nice_date [gof lastLogon] ]
        set enabled [expr $uac&amp;amp;0x2 ]
        set enabstr "False"
       if {$enabled} {set enabstr "True"}
       puts $name$sep$sam$sep$exp$sep$locked$sep$lastlogon$sep$enabstr
}
```

user-report

Use this command to lists the users in the specified zone. The command line arguments are as follows:

-z	required	The distinguished name of the zone
-m	optional	Bind using the ADEdit host machine's credentials (see bind) Note: If you use -m you do not need to enter -u
-u	optional	Administrator's account name.
-p	optional	Administrator's account password. Note: If you do not enter the password in the command line you will be prompted to enter it.

```
#!/bin/env adedit
# This script lists the users in the zone you specify in the command line.
# On the command line use either -m or -u
package require ade_lib
proc usage {msg} {
        puts {usage: -z <zoneDN> [-m] [-u <user>] [-p <password>]}
```

```
            puts $msg
            exit 1
}
if {[getopt argv -z zoneDN] == 0} {
            usage "Missing input zone. Enter full distinguished name"
}
if {[catch {domain_from_dn $zoneDN} domain]} {
            usage "Invalid input zone name. Enter full distinguished name"
}
set verbose 0
if {[getopt argv -v]} {
            set verbose 1
}
if {[getopt argv -m]} {
            bind -machine $domain
} else {
   if {[getopt argv -u user]} {
      if {[getopt argv -p password]} {
            bind $domain $user $password
       } else {
        bind $domain $user}
             } else {
    bind $domain
  }
}
select_zone $zoneDN
list_zone_users
```

GetComputers

Use this command to list all the Centrify-managed computers in the specified domain. Enter the domain name in the command line.

```
#!/bin/env adedit
# GetComputers
# Purpose: Retrieves a listing of all UNIX computers in all DelineaZones.
package require ade_lib
puts "
This script retrieves a listing of all UNIX computers in the specified domain"
puts "and shows the zone to which it is joined"
if { $argc == 0 } {
            puts "
            Command format: $argv0 domain name"
            exit 1
 }
set domain [lindex $argv 0]
# Use lindex command because argv is a list and bind requires a string
puts "
Enter administrator name for bind command"
gets stdin admin
bind $domain $admin
foreach ZONE [get_zones $domain] {
            select_zone $ZONE
            foreach COMPUTER [get_zone_computers] {
                        puts -nonewline $COMPUTER:; puts $ZONE;
            }
}
```

Run a Script from a Script

The following scripts illustrate the use of the Tcl source command to run the script in a specified file. In this example, the source file is setenv, which prompts the user to enter environment variables such as the domain and zone.

> **Note** You may find repeated use of setenv to be maddening since it prompts you for all of the environment variables regardless of whether the command actually needs them. This is done for demonstration purposes only. In a production environment, you would eliminate the prompts you don't need by tailoring setenv specifically to your environment. Feel free to remove or comment out parts when you've had enough.

The subsequent scripts in this section call the setenv script and then run a short script that does simple queries, such as get the child zones, get the computers in the zone, and get the groups.

setenv

This script prompts you to enter data that can be used in the calling script. This example is intended as a demonstration only. Not all of the information is relevant to the calling script. Note that this sample script assumes you are using the default deployment structure with a top-level organizational unit. If you are not using the default deployment structure, you should modify the sample script to reflect the structure you are using before testing its operation.

```
# Setenv file contents
```

```
# Purpose: Sets up a common environment for the following Active Directory
# tools, selecting the Active Directory Domain, binding the user, and
# defining commonly used variables.
# Other Active Directory tools:
# GetZones
# GetUsers
# GetGroups
# GetChildZones
# GetComputers
puts "
This portion of the script prompts you to enter the domain and account name for the bind command."
# If you are always using the same domain, comment out the puts and gets and use the set command instead
puts "
Enter the domain name"
gets stdin domain
# get the distinguished name for the domain.
set domaindn [dn_from_domain $domain]
puts "
Enter administrator account name for bind command"
gets stdin admin
bind $domain $admin
puts "
bind to $domain complete"
puts "
The next two prompts ask you to enter the OU and container for your zone information"
puts "
Enter the name of the Active Directory container that holds the Delineazone-related data"
gets stdin zonesContainer
# If you are always using the same zone, comment out the puts and gets and use the set command instead
# set zonesContainer <Active Directory container with zones data>
puts "
Enter the name of the organizational unit that has the zone container."
gets stdin zonesContainerOU
# If you are always using the same OU for the zone container, comment out the puts and gets and use the set command instead
# set zonesContainerOU <Active Directory OU with zones container>
puts "
Enter the base organizational unit with the Delineamanaged computers data"
gets stdin baseOU
# If you are always using the same base OU, comment out the puts and gets commands and use the set command instead
# set baseOU <base OU name>
puts "
The next prompt asks for the parent zone."
# If you are always using the same zone, comment out the puts and gets and use the set command instead
# set parentZone <parent zone name>
puts "
Enter the parent zone name"
gets stdin parentZone
```

GetZones

Use this script to get a list of all the zones in a domain.

```
#!/bin/env adedit
# GetZones
# Purpose: Performs a recursive listing of all Delineazones in the specified
# domain
package require ade_lib
source setenv
puts "
This script retrieves a recursive listing of all Delineazones in the $domain domain"
puts "
The Active Directory folder with the Delineazone data is named $zonesContainer"
puts "
That container is in organizational unit $zonesContainerOU"
puts "
The parent zone is $parentZone"
foreach ZONE [get_zones $domain] {
        puts $ZONE;
}
```

GetUsers

Use this script to get a list of all users in a zone.

```
# GetUsers
# Purpose: Operates on a recursive listing of all UNIX users in all
# DelineaZones, and retrieves the administered UNIX attribute values
# for each user object in each zone.
package require ade_lib
puts "
This script retrieves the UNIX attributes for each user in each zone in the specified domain"
source setenv
```

```
foreach ZONE [get_zones $domain] {
    select_zone $ZONE
    foreach USER [get_zone_users] {
        save_zone_user $USER
        puts  -nonewline "[get_zone_user_field uname]:[gzuf uid]:[gzuf gid]:[gzuf gecos]:[gzuf home]:[gzuf shell]"; puts :$USER:$ZONE
    }
}
```

GetGroups

Use this script to get the UNIX group attribute values for the groups in the managed computers.

```
#!/usr/bin/env adedit
# GetGroups
# Purpose: Retrieves the UNIX group attribute values for each UNIX
# group administered in the parent zone specified in setenv.
# To select a different zone, change the DN in the select_zone command
package require ade_lib
puts "
This script retrieves the group attribute values for each UNIX group in the specified parent zone"
source setenv
select_zone "CN=$parentZone,CN=$zonesContainer,OU=$zonesContainerOU,$domaindn"
foreach GROUP [get_zone_groups] {
    select_zone_group $GROUP
    puts  -nonewline "[get_zone_group_field name]:[gzgf gid]"; puts :$GROUP
}
```

GetChildZones

Use this command to get a list of the child zones for the specified parent.

```
#!/bin/env adedit
# # GetChildZones
# Purpose: Retrieves a recursive listing of all new hierarchical Delineachild
# zones administered underneath the parent zone specified in setenv
#
package require ade_lib
source setenv
puts "
This script retrieves a recursive listing of all child zones in $parentZone"
puts "
The Active Directory folder with the Delineazone information is $zonesContainer"
select_zone "CN=$parentZone,CN=$zonesContainer,OU=$zonesContainerOU,$domaindn"
foreach ZONE [get_child_zones -tree] {
    puts $ZONE;
}
```

ADEdit Command Reference

This chapter describes each ADEdit command in alphabetical order. Each command description includes details about the options and arguments you can specify and the values returned, if applicable.

Inn, addition, some ADEdit commands can only be used when you are working with hierarchical zones. Other commands can be used in classic or hierarchical zones, but require you to specify the zone type. For each command, the **Zone type** section indicates whether there are any zonerelated constraints as follows:

- **Hierarchical only**: You must have a hierarchical zone selected for the command to work.
- **Classic and hierarchical**: You can use the command in both classic zones and hierarchical zones. Options in the command let you specify whether you are working with a classic or hierarchical zone. In most cases, commands that work in both classic and hierarchical zones, require the classic zone to be a classic4 zone. The classic3 zone type is intended for backward compatibility with older agents and only commands where the zone type is not applicable are supported.
- **Classic only**: You must have a classic4 zone selected for the command to work.
- **Not applicable**: You can use the command because the zone type does not matter.

In addition to the zone type, syntax, and return values, each command description includes at least one usage example and a summary of related commands, if appropriate.

add_command_to_role

Use the add_command_to_role command to add a privileged UNIX command to the currently selected role that is stored in memory. The command must already exist. You can create privileged UNIX commands using new_dz_command.

The add_command_to_role command does *not* change the role as it is stored Active Directory. Running the command changes the role only in memory. You must save the role before the added command takes effect in Active Directory. If you select another role or quit ADEdit before saving the role, any commands you've added since the last save won't take effect.

Zone Type

Classic and hierarchical

Syntax

add_command_to_role command[/zonename]

Abbreviation

acr

Options

This command takes no options.

Arguments

This command takes the following argument:

command[*/zonename*]	string	Required. Specifies the name of an existing UNIX command to add to the currently selected role. If the UNIX command right that you want to add is defined in the current zone, the *zonename* argument is optional. If the UNIX command right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific UNIX command right to add.

Return Value

This command returns nothing if it runs successfully.

Examples

```
add_command_to_role basicshell/global
```

This example adds the command basicshell, defined in the global zone, to the currently selected role.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select a role to work with:

- get_role_commands returns a Tcl list of the UNIX commands for the role.
- new_role creates a new role.
- select_role retrieves a role from Active Directory.

The following commands enable you to work with a currently selected role:

- add_pamapp_to_role adds a PAM application to the role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications for the role.
- get_role_field reads a field value from the role.
- list_role_rights lists of all privileged commands and PAM application rights for the role.
- remove_command_from_role removes a UNIX command from the role.
- remove_pamapp_from_role removes a PAM application from the role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the role.

add_map_entry

Use the add_map_entry command to add an entry to the currently selected NIS map stored in memory. This command does not support a comment field. If you want to add a comment along with the entry use add_map_entry_with_comment instead.

To change an existing entry in a NIS map, use delete_map_entry to remove the entry, then add the revised version using add_map_entry.

The add_map_entry` command changes the NIS map in memory and in Active Directory. You do not need to save the NIS map for the added entry to take effect in Active Directory.

Zone Type

Not applicable

Syntax

```
add_map_entry key value
```

Abbreviation

```
ame
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

key	string	Required. Specifies the key of the NIS map entry.
value	string	Required. Specifies the value of the NIS map entry.

Return Value

This command returns nothing if it runs successfully.

Example

add_map_entry Finance “Hank@acme.com,Sue@acme.com”

This example adds the NIS map entry Finance with the value Hank@acme.com,Sue@acme.com to the currently selected NIS map.

Related Commands

The following commands enable you to view and select the NIS map you want to work with:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout of all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use additional commands to work with that map's entries or use the following commands to delete or save the currently selected NIS map:

- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

add_map_entry_with_comment

Use the add_map_entry_with_comment command to add an entry to the currently selected NIS map stored in memory and lets you include a comment. The comment can be up to 2048 characters and does not support new line syntax.

To change an existing entry in a NIS map, use delete_map_entry to remove the entry, then add the revised version using add_map_entry_with_comment.

The add_map_entry_with_comment command changes the NIS map in memory and in Active Directory. You do not need to save the NIS map for the added entry to take effect in Active Directory.

Zone Type

Not applicable

Syntax

add_map_entry_with_comment key value comment

Abbreviation

amewc

Options

This command takes no options.

Arguments

This command takes the following arguments:

key	string	Required. Specifies the key of the NIS map entry.

value	string	Required. Specifies the value of the NIS map entry.
comment	string	Required. Specifies the comment for the NIS map entry.

Return Value

This command returns nothing if it runs successfully.

Example

```
add_map_entry_with_comment Finance “Hank@acme.com,Sue@acme.com” “new Finance staff”
```

This example adds the NIS map entry Finance, with the value Hank@acme.com,Sue@acme.com and comment new Finance staffto the currently selected NIS map.

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select a NIS map to work with:

- get_nis_maps returns a Tcl list of NIS maps in the current zone.
- list_nis_maps lists to stdout the NIS maps in the current zone.
- new_nis_map creates a new NIS map.
- select_nis_map retrieves a NIS map from Active Directory.

The following commands enable you to work with a currently selected NIS map:

- add_map_entry adds an entry to the NIS map.
- delete_map_entry removes an entry from the NIS map.
- get_nis_map_field reads a field value from the NIS map.
- get_nis_map and get_nis_map_with_comment return a Tcl list of NIS map entries.
- list_nis_map and list_nis_map_with_comment lists NIS map entries to stdout.

add_object_value

Use the add_object_value command to add a value to a multi-valued field (attribute) of a specified Active Directory object in Active Directory. This command only works on the object in Active Directory, not on the currently selected Active Directory object in memory (if there is one).

If the added value isn't valid, Active Directory will report an error and add_object_value won't save the value.

This command is useful for fields that may be very large—members of a group, for example.

Zone Type

Not applicable

Syntax

```
add_object_value dn field value
```

Abbreviation

aov

Options

This command takes no options.

Arguments

This command takes the following arguments:

dn	string	Required. Specifies the distinguished name (DN) of the Active Directory object in which to add a value.
field	string	Required. Specifies the name of a multi-valued field in the currently selected Active Directory object to which to add the value. This can be any field that is valid for the type of the currently selected Active Directory object.
value		Required. Specifies the value to add to the field. The type of value depends on the field specified by the *field* argument.

Return Value

This command returns nothing if it runs successfully.

Examples

```
add_object_value cn=groups,dc=acme,dc=com users adam.avery
```

This example adds the value adam.avery to the users field of the groups object specified by the DN.

Related Commands

The following commands enable you to work with Active Directory objects:

- delete_object deletes the Active Directory object from Active Directory.
- delete_sub_tree deletes the Active Directory object and all of its children.
- get_object_field reads a field value from the Active Directory object.
- remove_object_value removes a value from a multi-valued attribute of the Active Directory object.
- save_object saves the Active Directory object.
- set_object_field sets a field value in the Active Directory object.

add_pamapp_to_role

Use the add_pamapp_to_role command to add a PAM application right to the currently selected role stored in memory. The PAM application right must already exist. You can create PAM application rights using new_pam_app.

The add_pamapp_to_role command does *not* change the role as it is stored Active Directory. The command only changes the role stored in memory. You must save the role using save_role before the added PAM application takes effect in Active Directory. If you select another role or quit ADEdit before saving the role, any PAM application rights you've added since the last save won't take effect.

You can only use the add_pamapp_to_role if the currently selected zone is a classic4 or hierarchical zones. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
add_pamapp_to_role app[/zonename]
```

Abbreviation

```
apr
```

Options

This command takes no options.

Arguments

This command takes the following argument:

app[*/zonename*]	string	Required. Specifies the name of an existing PAM application right to add to the currently selected role. If the PAM application right that you want to add is defined in the current zone, the *zonename* argument is optional. If the PAM application right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific PAM application right to add.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example adds the PAM application login-all, which is defined in the currently selected zone, to the currently selected role:

```
add_pamapp_to_role login-all
```

The following example adds the PAM application access right oracle-admin from the emea zone to the currently selected role:

```
add_pamapp_to_role oracle-admin/emea
```

Related Commands

The following commands enable you to view and select the role you want to work with:

- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.
- get_roles returns a Tcl list of roles in the current zone.
- list_roles displays a list to stdout of all roles in the currently selected zone.

After you have a role stored in memory, you can use additional commands to work with that role's fields, commands, and applications or use the following commands to delete or save the currently selected role:

- save_role saves the selected role with its current settings to Active Directory.
- delete_role deletes the selected role from Active Directory and from memory.

add_sd_ace

Use the add_sd_ace command to add an access control entry (ACE) in ACE string form to a security descriptor (SD) in SDDL (security descriptor description language) form.

The command takes an ACE string and an SDDL string. The command writes the ACE string there. The command returns an SDDL string that includes the added ACE string.

Zone Type

Not applicable

Syntax

```
add_sd_ace sddl_string ace_string
```

Abbreviation

```
ase
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

sddl_string	string	Required. Specifies a security descriptor in SDDL format.
ace_string	string	Required. Specifies an access control entry in ACE string form (which is always enclosed in parentheses)

Return Value

This command returns a security descriptor string in SDDL format if it runs successfully.

Examples

This example adds an ACE string (A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY) to an SDDL at the end of the command.

```
add_sd_ace O:DAG:DAD:AI(A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA)(OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO)
(OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO)(OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO)
(A;;RCLCRPLO;;;AU)(OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015e5f28;;AO)
(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)(OA;CIIOID;RCLCRPLO;;bf967a9c-
0de6-11d0-a285-00aa003049e2;RU)(OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS)(A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA)
(A;CIID;LC;;;RU)(A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA) (A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY)
```

This example returns:

```
O:DAG:DAD:AI(A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY)(A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA)
(OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO)(OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO)
(OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO)(A;;RCLCRPLO;;;AU)(OA;;CCDC;4828cc14-1437-45bc-
9b07-ad6f015e5f28;;AO)(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-
00aa003049e2;RU)(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-
00aa003049e2;RU)(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-
00aa003049e2;RU)(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-
00aa003049e2;RU)(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-
00aa003049e2;RU)(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-
00aa003049e2;ED)(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-
00aa003049e2;ED)(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-
00aa003049e2;ED)(OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)(OA;CIIOID;RCLCRPLO;;
bf967a9c-0de6-11d0-a285-00aa003049e2;RU)(OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS)(A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA)
(A;CIID;LC;;;RU)(A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA)
```

Related Commands

The following commands enable you to work with security descriptor strings:

- explain_sd converts security descriptor in SDDL format to a human-readable form.
- remove_sd_ace removes an access control entry (ACE) from a security descriptor.
- set_sd_owner sets the owner of a security descriptor.

bind

Use the bind command to bind ADEdit to a domain. Multiple bind commands can bind ADEdit to multiple domains in multiple forests. ADEdit must be bound to at least one domain before its commands have any effect on Active Directory or Delineaobjects. When ADEdit is bound to multiple domains, its commands can work on any of those domains.

You can use bind to bind to any domain for which the DNS can resolve a name and for which you have log-in permission. ADEdit's host computer does not need to be joined to a domain for ADEdit to bind to and work on that domain.

You can optionally specify a server in the domain to bind to, in which case ADEdit binds to that domain controller. If you don't specify a server, ADEdit automatically binds to the closest, fastest domain controller. You can use options to request automatic binding to a global catalog (GC) domain controller or to a writable domain controller.

You can authorize the bind connection to a domain controller in the following ways:

- If you provide no user or password arguments, bind uses the user name and password stored in the current Kerberos credential cache on the ADEdit host computer.
- If you provide a user argument without the password argument, bind in interactive mode prompts you for a password, then uses the user argument along with your entered password for authorization.
- If you provide a user argument and password argument, bind uses the user and password arguments for authorization.
- If you specify the -machine option, ADEdit authenticates using the credentials for the ADEdit host computer. You cannot provide user or password arguments if you specify the -machine option. Note that you must have read permission on the host's credential files to use this option, so you must typically have root permissions to use the option.

Zone Type

Not applicable

Syntax

bind [-gc] [-write] [-machine] [*server*@]*domain* [*user* [*password*]]

Abbreviation

None

Options

This command takes the following options:

-gc	Requests an automatic binding to a global catalog (GC) domain controller. This option has no effect if there's a domain controller specified using the *server* argument.
-write	Requests an automatic binding to a writable domain controller. This option has no effect if there's a domain controller specified using the *server* argument.
-machine	Binds using the credentials for the ADEdit host computer. Note that most computer accounts have only read permission, not write permission for Active Directory. To use this option, you must have read permission on the local computer's keytab file and credentials cache. In most cases, only the root user has this right.

Arguments

This command takes the following arguments:

[*server*]@domain	string	Required. Specifies the domain to bind to. If you want to specify a domain controller to connect to, precede the domain with the name of the domain controller's server followed by the "@" symbol. If you don't specify a domain controller, bind performs an automatic binding to the domain controller that ADEdit determines is most optimal for binding.

[*user*]	string	Optional. Specifies the user name for logging on to the domain controller. If you don't specify this argument and the -machine option is also not present, ADEdit attempts to log on using your current account credentials. If you specify the -machine option, you cannot use this argument.
[*password*]	string	Optional. Requires the *user* argument. Specifies the password to use when logging on to the domain controller as *user*.

Return Value

This command returns no value.

Examples

The following example binds ADEdit to the domain acme.com, logging in as administrator with the password #3gEgh^&4:

```
bind acme.com administrator #3gEgh^&4
```

Note that a password that includes Tcl-special characters such as $ might trigger character substitution that modifies the password. To ensure that a password isn't altered by the Tcl interpreter, enclose the password in braces {}. For example:

```
bind acme.com maya,garcia {$m1l3s88}
```

Related Commands

The following commands perform actions related to the bind command:

- get_bind_info returns information about a domain to which ADEdit is bound.
- pop restores the context from the top of ADEdit's context stack to ADEdit.
- push saves ADEdit's current context to ADEdit's context stack.
- show returns the current context of ADEdit: its bound domains and its currently selected objects.

clear_rs_env_from_role

Use the clear_rs_env_from_role command to remove the restricted shell environment from the currently selected role that is stored in memory.

The clear_rs_env_from_role command does not modify the information stored in Active Directory for the role. If you run this command using ADEdit without saving the role to Active Directory, the change will have no effect on the restricted shell environment stored in Active Directory.

You can only use the clear_rs_env_from_role command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

```
clear_rs_env_from_role
```

Abbreviation

```
crse
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
clear_rs_env_from_role
```

This example removes the restricted shell environment from the current role.

Related Commands

The following commands perform actions related to this command:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.
- set_rs_env_for_role assigns a restricted shell environment to the current role.

After you have a restricted shell environment stored in memory, you can use the following commands to work with that: restricted shell environment:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.

create_computer_role

Use the create_computer_role command to create a new computer role in Active Directory. The command does *not* store the new computer role in memory nor set it as the currently selected ADEdit computer role. To manage the computer role, you must select it using select_zone and then use zone commands to work with the computer role's fields.

ADEdit requires a valid license before the computer role is created. The create_computer_role command does an implicit search. The first place it looks is the ADEdit context for a valid license indicator (see the validate_license command) for the forest. If an indicator is not in the context, the command checks for a valid license as follows:

- Bind to the global catalog (GC) domain controller, search the forest for the license container and validate the license.
- Bind to the current domain, search for the license container and validate the license.

If it finds a valid license, it stores an indicator in the current context and creates the new computer role. If it does not find a valid license, create_computer_role reports "No valid license found" and exits. If the command fails, use validate_license to validate the license container explicitly.

To associate role assignments with the new computer role, you must select the computer role, then use new_role_assignment.

Zone Type

Hierarchical only

Syntax

```
create_computer_role computer_role_path group_upn
```

Abbreviation

```
ccr
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

computer_role_path	string	Required. Specifies a path to the new computer role. The path consists of the hosting zone's distinguished name followed by a slash and the name of the new computer role.
group_upn	string	Required. Specifies the user principal name (UPN) of a computer group in Active Directory to associate with this computer role. This computer group defines the set of computers in which this computer role functions. The computer group must be available within the computer role's host domain.

Return Value

This command returns no value if it runs successfully.

Examples

The following example creates a new computer role named LinuxComputers in the global zone of acme.com:

```
create_computer_role {CN=global,CN=Zones,CN=Acme,DC=acme,DC=com/LinuxComputers} linux_computers@acme.com
```

The scope of the computer role is defined by the group named linux_computers which is an Active Directory groups defined in acme.com. To work with the new computer role, you must select it as a zone:

```
select_zone "CN=global,CN=Zones,CN=Acme,DC=acme,DC=com/LinuxComputers"
```

Related Commands

The following command retrieves the computer role from Active Directory and stores it in memory so you can use other commands to work with it.

- select_zone retrieves the computer role and stores it in memory.

After you have a computer role selected as a zone, you can use the following commands to view and manage the computer role:

- new_role_assignment creates a new role assignment for the selected computer role.
- list_role_assignments lists user role assignments for the selected computer role.
- get_role_assignments returns a Tcl list of user role assignments for the selected computer role.
- get_zone_field retrieves what computer group is associated with the computer role.
- set_zone_fieldsets what computer group is associated with the computer role.
- delete_zone deletes the selected computer role from Active Directory and memory.

create_zone

Use the create_zone command to create a new zone in Active Directory. The command does *not* store the new zone in memory nor set it as the currently selected ADEdit zone. To manage the zone, you must select it using select_zone and then use zone commands.

This command can create different types of zones and the zones can use different types of schemas, depending on the schema you are using for Active Directory. Before the zone is created, however, ADEdit checks for a valid license.

The create_zone command first checks the ADEdit context for a valid license indicator for the forest. If an indicator is not found in the context, the command checks for a valid license as follows:

- Bind to the global catalog (GC) domain controller, search the forest for the license container and validate the license.
- Bind to the current domain, search for the license container and validate the license.

If the command finds a valid license, it stores an indicator in the current context and creates the new zone. If it does not find a valid license, create_zone reports "No valid license found" and exits. If the command fails, use the validate_license command to validate the license container explicitly.

> **Note**: When this command creates a zone, the zone contains predefined roles such as "sftp" and "UNIX Login." The zone does not, however, contain the role "Windows Login" because ADEdit does not support Windows rights.

Zone Type

Classic and hierarchical

Syntax

create_zone [-ou] [-nonisserversgroup] [-notdelegateanyright] zone_type path [schema_type]

Abbreviation

cz

Options

This command takes the following options:

-nonisserversgroup	Creates the new zone without the zone_nis_servers group.
-notdelegateanyright	Creates the new zone but does not set the zone permissions. If you use this option, be sure to set the zone permissions later.
-ou	Creates the new zone as an organizational unit object. If not present, the new zone is created as a container object. Note that the parent container determines what type of object the zone can be. If the parent container is a generic container object, the zone must be a container object. If the parent container is an organizational unit object, the zone can be either an organizational unit object or a container object.

Arguments

This command takes the following arguments:

zone_type	string	Required. The possible values are: **tree** specifies a hierarchical zone that can be a parent or child zone. **classic3** specifies a classic zone that is compatible with agent version 3 and later. **classic4** specifies a classic zone that is compatible with agent version 4 and later. **computer** specifies a computer-level zone that consists of a single computer in a hierarchical zone. This zone type is used to support computer-level overrides for user and group profiles and role assignments. It is not applicable in classic zones. **classic-computer** specifies a computer-level zone that consists of a single computer in a classic zone. This zone type is used to enable you to assign a role to a specific computer in classic zones. It is not applicable in hierarchical zones.
path	string	Required. Specifies a path to the new zone. The path consists of the new zone's distinguished name (DN) and (if a computer override) the name of the computer.
schema_type	string	Optional. Specifies the type of schema to use for the new zone. The possible values are: **sfu** specifies the Microsoft Services For UNIX schema. This setting can be used for tree, classic3, and classic4 zone types. If it's used for a hierarchical zone, it can only be the root of the zone hierarchy. **std** specifies the dynamic schema. This setting can be used for all zone types. This is the default schema unless ADEdit detects the RFC2307 schema. **rfc** specifies the RFC2307 schema.This setting can be used for all zone types. This is the default schema if ADEdit detects that RFC2307 is installed and the domain is at Windows Server 2003 functional level. If none of these values is present, the default is either **std** or **rfc** as described above.

Return Value

This command returns no value if it runs successfully.

Examples

The following examples illustrate how to create a classic zone, hierarchical zone, and computer-specific zone in Server Suite 2012 and later.

Classic Zone

The following command creates a classic zone named finance in the Acme organizational unit in the acme.com domain that uses the dynamic schema (std):

```
create_zone classic4 “CN=finance,OU=Acme,DC=acme,DC=com” std
```

Hierarchical Zone

The following command creates a new hierarchical parent zone named finance in the Zones container in the Acme organizational unit in the acme.com domain:

```
create_zone tree “CN=finance,CN=Zones,OU=Acme,DC=acme,DC=com” std
```

To make the finance zone a child zone within a global zone already created in the same container, OU, and domain, you would next select finance to make it the currently selected zone, then use set_zone_field (szf) to specify the global zone as its parent, and the save finance. For example:

```
select_zone “CN=finance,CN=Zones,OU=UNIX,DC=acme,DC=com”  szf parent “CN=global,CN=Zones,OU=UNIX,DC=acme,DC=com”  save_zone
```

Computer-specific Zone

The following command creates a computer-specific zone for the computer srv1 in the apache zone, which is a child of the global zone, in the Zones container in the Acme organizational unit in the acme.com domain.

```
create_zone computer svr1.acme.com@CN=apache,CN=global,CN=Zones,OU=Acme,DC=acme,DC=com
```

Related Commands

Before you use this command, you must bind to one or more Active Directory domains. The following command enables you to store a newly created zone in memory:

- select_zone retrieves a zone from Active Directory and stores it in memory.

After you have created a new zone and stored it in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

delegate_zone_right

Use the delegate_zone_right command to delegate an administrative right for the currently selected zone to a security principal (user or group). Zone rights allow a user or group to use and manage zone properties, including computer-specific zone properties and computer roles.

Zone Type

Classic and hierarchical

Syntax

```
delegate_zone_right right principal_upn
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following arguments:

right	string	Required. Specifies the right to delegate. Possible values: **add_computer_role**: The right to add computer roles to the zone. **add_computer_zone**: The right to add computer-specific zones. **add_group**: The right to add groups to the zone.
		add_nismap: The right to add NIS maps to the zone. **add_remove_nismap_entry**: The right to add or remove NIS map entries. **add_user**: The right to add users to the zone.
		add_user_group_to_computer_zone: The right to add user and group overrides to the selected computer-specific zone. **change_user**: The right to modify user profiles in the zone. **change_group**: The right to modify group profiles in the zone.
		change_computer: The right to modify computer profiles in the zone. **change_zone**: The right to change zone properties. **delegate_permission_for_computer_zone**: The right to delegate permissions to other users for computer-specific zones.
right (continued)	string (continued)	**delete_computer**: The right to remove computers from the zone. **delete_computer_role**: The right to delete computer roles in the zone. **delete_computer_zone**: The right to delete computer-specific zones.
		delete_group: The right to remove groups from the zone. **delete_user**: The right to remove users from the zone. **delete_user_group_from_computer_zone**: The right to delete user and group overrides from the selected computer-specific zone.
		delete_zone: The right to remove the zone. **enable_dz**: The right to initialize authorization (privilege elevation service) data. This right is only applicable in classic zones. **import**: The right to import users and groups into the zone.
		join: The right to join computers to the zone. **manage_role_assignments**: The right to modify the roles assigned in zones, computer-specific zones, and computer roles. **manage_roles_and_rights**: The right to modify role definitions and access rights.
		modify_computer_role: The right to modify computer role entries. This right is not applicable in classic zones. **modify_nismap_entry**: The right to modify NIS map entries. **modify_user_group_in_computer_zone**: The right to modify user and group overrides in the selected computer-specific zone.
right (continued)	string (continued)	**nisservers**: The right to allow computers to respond to NIS client requests. **remove_nismap**: The right to remove NIS maps.
principal_upn	string	Required. Specifies the user principal name (UPN) of a user or group in Active Directory to delegate the specified right to.

Return Value

This command returns no value if it runs successfully.

Examples

```
delegate_zone_right add_user adam.avery@acme.com
```

This example delegates the right to add users to the currently selected zone to the Active Directory user Adam Avery.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone to work with:

- `create_zone` creates a new zone in Active Directory.
- `get_zones` returns a Tcl list of all zones within a specified domain.
- `select_zone` retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

delete_dz_command

Use the `delete_dz_command command to delete the currently selected privileged command from Active Directory and from memory. You cannot use other commands to manage privileged commands after deletion because there will be no currently selected command in memory.

Zone Type

Classic and hierarchical

Syntax

delete_dz_command

Abbreviation

dldzc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_dz_command

This example deletes the currently selected command from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command to work with:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- get_dzc_field reads a field value from the currently selected command.
- save_dz_command saves the selected command with its current settings to Active Directory.
- set_dzc_field sets a field value in the currently selected command.

delete_local_group_profile

Use the delete_local_group_profile command to delete a local UNIX or Linux group that has a profile defined in the current zone. When you delete a group, the group's zone object is deleted, but the group's entry in the local /etc/group file is retained.

Zone Type

Hierarchical only.

Syntax

delete_local_group_profile group_name

Abbreviation

dllgp

Options

This command takes no options.

Arguments

This command takes the following argument:

group_name	string	Required. Specifies the UNIX name of the local group to delete from the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_local_group_profile marketing

This example deletes the zone object for the local group marketing. The entry for marketing in the local /etc/group file is retained.

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

delete_local_user_profile

Use the delete_local_user_profile command to delete a local UNIX or Linux user that has a profile defined in the current zone. When you delete a user, the user's zone object is deleted, but the user's entry in the local /etc/passwd file is retained.

Zone Type

Hierarchical only.

Syntax

```
delete_local_user_profile user_name
```

Abbreviation

```
dllup
```

Options

This command takes no options.

Arguments

This command takes the following argument:

user_name	string	Required. Specifies the UNIX name of the local user to delete from the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_local_user_profile anton.splieth
```

This example deletes the zone object for the local user anton.splieth. The entry for anton.splieth in the local /etc/passwd file is retained.

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.

- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

delete_map_entry

Use the delete_map_entry command to delete an entry from the currently selected NIS map stored in memory. The delete_map_entry command changes the NIS map in memory and in Active Directory. You do not need to save the NIS map for the deleted entry to take effect in Active Directory.

Zone Type

Not applicable

Syntax

delete_map_entry key:index

Abbreviation

dlme

Options

This command takes no options.

Arguments

This command takes the following argument:

key:index	string	Required. Specifies the key of the NIS map entry to delete followed by a colon : and the index number of the key.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_map_entry calla:1

This example deletes the NIS map entry with the key value "calla" and index number 1 from the currently selected NIS map.

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select the NIS map to work with:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout all of the NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map's entries:

- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout the map entries in the currently selected NIS map.
- add_map_entry or add_map_entry_with_comment adds an map entry to the currently selected NIS map.

delete_nis_map

Use the delete_nis_map command to delete the currently selected NIS map from Active Directory and from memory. You cannot use other commands to

manage the NIS map after deletion because there will be no currently selected map in memory.

Zone Type

Not applicable

Syntax

delete_nis_map

Abbreviation

dlnm

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_nis_map

This example deletes the currently selected NIS map from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select the NIS map to work with:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout of all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map's entries:

- add_map_entry or add_map_entry_with_comment adds an entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout of the entries in the currently selected NIS map.

delete_object

Use the delete_object command to delete the currently selected Active Directory object from Active Directory and from memory. You cannot use other commands to manage the object after deletion because there will be no currently selected Active Directory object in memory.

> **Note**:Do NOT use the delete_object command to delete an Active Directory user or group that has been provisioned. If you use delete_object to delete a provisioned user or group, you create orphan user or group UNIX data objects. Instead, use the delete_zone_user or delete_zone_group command. In addition, you would use the select_zone_user and select_zone_group rather than select_object to select the user or group. For information about displaying orphan accounts, see the list_zone_users and list_zone_groups.

Zone Type

Not applicable

Syntax

delete_object

Abbreviation

dlo

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_object

This example deletes the currently selected Active Directory object from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected Active Directory object stored in memory. The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of matching objects.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object with its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use other commands to work with that object's attributes, or the following commands to delete or save information for the object:

- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- save_object saves the selected Active Directory object with its current settings to Active Directory.

delete_pam_app

Use the delete_pam_app command to delete the currently selected PAM application from Active Directory and from memory. You cannot use other commands to manage the PAM application after deletion because there will be no currently selected PAM application in memory.

Zone Type

Classic and hierarchical

Syntax

delete_pam_app

Abbreviation

dlpam

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_pam_app
```

This example deletes the currently selected PAM application from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. After you have a zone stored in memory, you can use the following commands to view and select the PAM application to work with:

- `get_pam_apps` returns a Tcl list of PAM application rights in the current zone.
- `list_pam_apps` lists to `stdout` all PAM application rights in the current zone.
- `new_pam_app` creates a new PAM application right and stores it in memory.
- `select_pam_app` retrieves a PAM application right from Active Directory and stores it in memory

After you have a PAM application stored in memory, you can use the following commands to work with that PAM application's attributes, delete the PAM application, or save information for the PAM application:

- `delete_pam_app` deletes the selected PAM application right from Active Directory and from memory.
- `get_pam_field` reads a field value from the currently selected PAM application right.
- `save_pam_app` saves the selected PAM application right with its current settings to Active Directory.
- `set_pam_field` sets a field value in the currently selected PAM application right.

delete_role

Use the `delete_role` command to delete the currently selected role from Active Directory and from memory. You cannot use other commands to manage the role after deletion because there will be no currently selected role in memory.

Zone Type

Classic and hierarchical

Syntax

```
delete_role
```

Abbreviation

```
dlr
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_role
```

This example deletes the currently selected role from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- `get_roles` returns a Tcl list of roles in the current zone.
- `list_roles` lists to `stdout` all roles in the currently selected zone.
- `new_role` creates a new role and stores it in memory.
- `select_role` retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- `add_command_to_role` adds a UNIX command to the currently selected role.
- `add_pamapp_to_role` adds a PAM application to the currently selected role.
- `get_role_apps` returns a Tcl list of the PAM applications associated with the role.
- `get_role_commands` returns a Tcl list of the UNIX commands associated with the role.
- `get_role_field` reads a field value from the currently selected role.
- `list_role_rights` lists to `stdout` all UNIX commands and PAM applications associated with the role.
- `remove_command_from_role` removes a UNIX command from the currently selected role.
- `remove_pamapp_from_role` removes a PAM application from the currently selected role.
- `save_role` saves the selected role with its current settings to Active Directory.
- `set_role_field` sets a field value in the currently selected role.

delete_role_assignment

Use the `delete_role_assignment` command to delete the currently selected role assignment from Active Directory and from memory. You cannot use other commands to manage the role assignment after deletion because there will be no currently selected role assignment in memory.

Zone Type

Classic and hierarchical

Syntax

```
delete_role_assignment
```

Abbreviation

```
dlra
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_role_assignment
```

This example deletes the currently selected role assignment from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected role assignment stored in memory. The following commands enable you to view and select the role assignment to work with:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- list_role_assignments lists to stdout all role assignments in the currently selected zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use other commands to work with that role assignment's fields, or the following commands to save information for the role assignment:

- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- write_role_assignment saves the selected role assignment to a file.

delete_rs_command

Use the delete_rs_command command to delete the currently selected restricted shell command from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for restricted shell commands because there will be no currently selected restricted shell command available in memory.

Zone Type

Classic only

Syntax

delete_rs_command

Abbreviation

dlrsc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_rs_command

This example deletes the currently selected restricted shell command from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.

- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

delete_rs_env

Use the delete_rs_env command to delete the currently selected restricted environment from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for a restricted shell environment because there will be no currently selected restricted shell environment available in memory.

Zone Type

Classic only

Syntax

delete_rs_env

Abbreviation

dlrse

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_rs_env

This example deletes the currently selected RSE from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

delete_sub_tree

Use the delete_sub_tree command to delete an object and all of its child objects from Active Directory. Only child objects that are in the same container as the specified parent object are deleted. Child objects in other containers are not deleted.

WARNING: This is a very powerful command, and can cause a lot of damage if used incorrectly. It's similar to running rm -rf * in UNIX.

In interactive mode, ADEdit prompts you for confirmation before executing this command. If you use this command in a script, ADEdit does not prompt for confirmation. You should use caution before using this command in a script.

This command can be used on any Active Directory object, including a container, OU, computer object, group or user. However, it is especially useful for deleting a corrupted zone. You'd normally use select_zone and then delete_zone to delete a zone. If the zone is damaged, though, select_zone might not work. In that case, delete_sub_tree will do the job.

If the zone is a hierarchical zone, this command deletes only the child zones in the same container as the parent zone. If there are any child zones in other containers, they are not deleted.

Zone Type

Classic and hierarchical

Syntax

```
delete_sub_tree dn
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following argument:

dn	DN	Required. Specifies the distinguished name of the object (with all of its children) to remove from Active Directory.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_sub_tree "CN=marketing,CN=Zones,CN=Acme,DC=acme,DC=com"
```

This example deletes the currently selected "marketing" zone with all of its children from Active Directory.

Related Commands

The following commands enable you to view and manage the Active Directory object to work with:

- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of matching objects.
- new_object creates a new Active Directory object and stores it in memory.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- select_object retrieves an object with its attributes from Active Directory and stores it in memory.

The following commands enable you to view and manage Active Directory object attributes:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- set_object_field sets a field (attribute) value in the currently selected Active Directory object.

delete_zone

Use the delete_zone command to delete the currently selected zone from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for zones because there will be no currently selected zone available in memory.

This command performs an LDAP sub-tree deletion operation. Only child zones that are in the same container as the specified parent zone are deleted. Child zones that are located in other containers are not deleted. Child zones that are based on pointers defined in the child zone are not deleted. For more information about deleting sub-tree objects, see delete_sub_tree.

In interactive mode, ADEdit prompts you for confirmation before executing this command. If you use this command in a script, ADEdit does not prompt for confirmation. You should use caution before using this command in a script.

Zone Type

Classic and hierarchical

Syntax

delete_zone

Abbreviation

dlz

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_zone

This example deletes the currently selected zone or computer role from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the zone to work with:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.
- select_zone retrieves a zone from Active Directory and stores it in memory as the currently selected zone.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates an administrative right to a specified user or group.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones for the current zone.
- get_zone_field reads a field value from the currently selected zone.
- set_zone_field sets a field value in the currently selected zone.

- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.

delete_zone_computer

Use the delete_zone_computer command to delete the currently selected zone computer profile from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for zone computer profiles because there will be no currently selected zone computer profile available in memory. This command only deletes the zone profile for the computer. It does not delete the Active Directory computer account.

Zone Type

Classic and hierarchical

Syntax

```
delete_zone_computer [-all]
```

Abbreviation

```
dlzc
```

Options

This command takes the following option:

-all	Removes the corresponding computer-specific zone profile if the selected computer is a computer-specific override zone.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_zone_computer
```

This example deletes the currently selected zone computer profile from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone computer profile stored in memory. The following commands enable you to view and select the zone computer profile to work with:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computer profiles in the current zone.
- list_zone_computers lists to stdout all zone computer profiles in the current zone.
- new_zone_computer creates a new zone computer profile and stores it in memory.
- select_zone_computer retrieves a zone computer profile from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- get_zone_computer_field reads a field value from the currently selected zone computer profile.
- set_zone_computer_field sets a field value in the currently selected zone computer profile.
- save_zone_computer saves the selected zone computer profile with its current settings to Active Directory.

delete_zone_group

Use the delete_zone_group command to delete the currently selected zone group profile from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for zone groups because there will be no currently selected zone group available in memory.

Zone Type

Classic and hierarchical

Syntax

```
delete_zone_group
```

Abbreviation

dlzg``

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
delete_zone_group
```

This example deletes the currently selected zone group from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone group stored in memory. The following commands enable you to view and select the zone group to work with:

- get_zone_groups returns a Tcl list of the Active Directory names of all zone groups in the current zone.
- list_zone_groups lists to stdout all zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- get_zone_group_field reads a field value from the currently selected zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

delete_zone_user

Use the delete_zone_user command to delete the currently selected zone user profile from Active Directory and from memory. After you run this command, you cannot run subsequent ADEdit commands for zone users because there will be no currently selected zone user available in memory.

Zone Type

Classic and hierarchical

Syntax

```
delete_zone_user
```

Abbreviation

dlzu

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

delete_zone_user

deletes the currently selected zone user from Active Directory and from memory.

Related Commands

Before you use this command, you must have a currently selected zone user stored in memory. The following commands enable you to view and select the zone user to work with:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone. * list_zone_users lists to stdout all zone users in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

dn_from_domain

Use the dn_from_domain command to convert a specified domain name in dotted form (acme.com, for example) to a distinguished name (DN). This conversion doesn't require lookup in Active Directory. The command performs a simple text conversion.

Zone Type

Not applicable

Syntax

dn_from_domain domain_name

Abbreviation

dnfd

Options

This command takes no options.

Arguments

This command takes the following argument:

domain_name	string	Required. Specifies a dotted domain name (acme.com, for example)

Return Value

This command returns a domain name as a distinguished name.

Examples

```
dn_from_domain acme.com
```

This example returns the domain name in this form: `dc=acme,dc=com`

Related Commands

The following commands convert information from one format to another:

- `domain_from_dn` converts a domain's distinguished name (DN) to a dotted name.
- `dn_to_principal` returns the sAMAccount@domain name or user principal name (UPN) for a security principal.

dn_to_principal

Use the `dn_to_principal` command to specify the distinguished name (DN) of a security principal (user, computer, or group). The command searches Active Directory for the principal, and if the principal is found, the command returns the `sAMAccount@domain` name of the principal. Optionally, you can also use this command to return the user principal name (UPN) for the principal.

Zone Type

Not applicable

Syntax

```
dn_to_principal [-upn] principal_dn
```

Abbreviation

```
dntp
```

Options

This command takes the following option:

-upn	Returns the principal name in user principal name (UPN) format, not the default sAMAccount@domain format.

Arguments

This command takes the following argument:

principal_dn	string	Required. Specifies the distinguished name (DN) of a security principal.

Return Value

This command returns the sAMAccount@domain name or (optionally) the user principal name (UPN) of a security principal. If the command doesn't find the specified security principal in Active Directory, it presents a message that it didn't find the principal.

Examples

```
dn_to_principal cn=brenda butler,cn=users,dc=acme,dc=com
```

This example returns: brenda.butler@acme.com

Related Commands

The following commands search for security principals in Active Directory:

- principal_to_dn searches Active Directory for a user principal name (UPN) and, if found, returns the corresponding distinguished name (DN).
- principal_from_sid searches Active Directory for an SID and returns the security principal associated with the SID.

domain_from_dn

Use the `domain_from_dn`` command takes a distinguished name (DN) that contains a domain and returns the domain name in dotted form (acme.com, for example). This conversion doesn't require lookup in Active Directory. The command performs a simple text conversion.

Zone Type

Not applicable

Syntax

```
domain_from_dn dn
```

Abbreviation

```
dfdn
```

Options

This command takes no options.

Arguments

This command takes the following argument:

dn	string	Required. Specifies a distinguished name that contains a domain.

Return Value

This command returns a domain name in dotted form such as acme.com. If the distinguished name doesn't contain domain component (DC) values, the command returns a notice that the DC values are missing.

Examples

```
dfdn cn=johndoe,cn=users,dc=acme,dc=com
```

This example returns: acme.com

Related Commands

The following command converts information from one format to another:

- dn_from_domain converts a domain's dotted name to a distinguished name.

explain_sd

Use the explain_sd command to specify a security descriptor (SD) in security descriptor description language (SDDL) form and returns a human-readable form of the security descriptor.

Zone Type

Not applicable

Syntax

```
explain_sd sddl_string
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following argument:

sddl_string	string	Required. Specifies a security descriptor in SDDL format.

Return Value

This command returns text that describes the supplied security descriptor in humanreadable form.

Examples

```
explain_sd O:DAG:DAD:AI(A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY)(A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA)
(OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO)(OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO)
(OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO)(A;;RCLCRPLO;;;AU)(OA;;CCDC;4828cc14-1437-45bc-
9b07-ad6f015e5f28;;AO)(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-
ad6f015e5f28;RU)(OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)
(OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED)
(OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU)(OA;CIIOID;RCLCRPLO;;bf967a9c-
0de6-11d0-a285-00aa003049e2;RU)(OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU)
(OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS)(A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA)
(A;CIID;LC;;;RU)(A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA)
```

This example returns the security descriptor information in readable form:

```
Owner: Domain Admins
Group: Domain Admins
Dacl: inherit supported,
 Allow | | delete,read SD,write DACL,change owner,create child,delete child,list children,self write,read property,write property,delete tree,list object,control access, | | | System
 Allow | | read SD,write DACL,change owner,create child,delete child,list children,self write,read property,write property,list object,control access, | | | Domain Admins
 Allow | | create child,delete child, | User | | Account operators
 Allow | | create child,delete child, | Group | | Account operators
 Allow | | create child,delete child, | Print-Queue | | Print operators
 Allow | | read SD,list children,read property,list object, | | | Authenticated users
 Allow | | create child,delete child, | inetOrgPerson | | Account operators
 Allow | inherit,inherit ony,inherited, | read property, | User-Account-Restrictions | inetOrgPerson | pre win2k
 Allow | inherit,inherit ony,inherited, | read property, | User-Account-Restrictions | User | pre win2k
 Allow | inherit,inherit ony,inherited, | read property, | User-Logon | inetOrgPerson | pre win2k
 Allow | inherit,inherit ony,inherited, | read property, | User-Logon | User | pre win2k
```

Allow | inherit,inherit ony,inherited, | read property, | Membership | inetOrgPerson | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | Membership | User | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | General-Information | inetOrgPerson | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | General-Information | User | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | RAS-Information | inetOrgPerson | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | RAS-Information | User | pre win2k
Allow | inherit,inherit ony,inherited, | read property, | Token-Groups | Computer | Enterprise Domain Controllers
Allow | inherit,inherit ony,inherited, | read property, | Token-Groups | Group | Enterprise Domain Controllers
Allow | inherit,inherit ony,inherited, | read property, | Token-Groups | User | Enterprise Domain Controllers
Allow | inherit,inherit ony,inherited, | read SD,list children,read property,list object, | | inetOrgPerson | pre win2k
Allow | inherit,inherit ony,inherited, | read SD,list children,read property,list object, | | Group | pre win2k
Allow | inherit,inherit ony,inherited, | read SD,list children,read property,list object, | | User | pre win2k
Allow | inherit,inherited, | read property,write property,control access, | Private-Information | | Self
Allow | inherit,inherited, | delete,read SD,write DACL,change owner,create child,delete child,list children,self write,read property,write property,delete tree,list object,control access, | | | Enterprise Admins
Allow | inherit,inherited, | list children, | | | pre win2k
Allow | inherit,inherited, | delete,read SD,write DACL,change owner,create child,list children,self write,read property,write property,list object,control access, | | | Administrators

Related Commands

The following commands enable you to work with security descriptor strings:

- remove_sd_ace removes an access control entry (ACE) from a security descriptor.
- add_sd_ace adds an access control entry to a security descriptor.
- set_sd_owner sets the owner of a security descriptor.

forest_from_domain

Use the forest_from_domain command to retrieve the forest name based on the domain name. The command also stores the retrieved forest name in memory.

Zone Type

Not applicable

Syntax

```
forest_from_domain [-nocache] <domain>
```

Abbreviation

ffd

Options

This command takes the following option:

nocache	Use this option to force fetch the forest name from Active Directory instead of reading the forest name from in memory.

Arguments

This command takes the following argument:

domain	string	Required. Specifies the domain for which you want to retrieve the forest.

Return Value

This command returns the forest name (in upper case text).

Examples

```
>forest_from_domain 5027f1d2.test
```

5027F1D1.TEST

>ffd 5027f1d2.test

5027F1D1.TEST

get_adinfo

Use the get_adinfo command to return information about the current join state for the ADEdit host computer. The command returns information about the joined domain, the joined zone, or the name the host computer is joined under.

Zone Type

Not applicable

Syntax

get_adinfo domain|zone|host

Abbreviation

adinfo

Options

This command takes no options.

Arguments

This command takes the following argument:

domain|zone|host	string	Required. The possible values are: **domain** returns the name of the currently joined domain. **zone** returns the distinguished name of the currently joined zone. **host** returns the name under which the ADEdit host computer is joined.

Return Value

This command returns a domain name, zone name, or computer name depending on the provided argument.

Examples

get_adinfo domain

This example returns the joined domain. For example: acme.com

get_adinfo zone

This example returns the path to the joined zone. For example:

CN=default,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com

Related Commands

None.

get_bind_info

Use the get_bind_info command to return information about one of ADEdit's currently bound domains. The command can return the name of the domain's forest, the name of the server bound within the domain, the security identifier (SID) of the domain, and the functional level of the domain or the domain's forest.

Zone Type

Not applicable

Syntax

get_bind_info domain forest|server|sid|domain_level|forest_level

Abbreviation

gbi

Options

This command takes no options.

Arguments

This command takes the following arguments:

domain	string	Required. Specifies the name of the domain for which to get information.
forest | server | sid | domain_level | forest_level	string	Required. The possible values are: **forest** returns the name of the forest that contains the bound domain. **server** returns the name of the domain server to which ADEdit is bound in the domain. **sid** returns the SID (security identifier) of the bound domain. **domain_level** returns the functional level of the bound domain, represented by an integer value: -1: unknown functional level 0: Windows 2000 Server 1: Windows Server 2003, interim level 2: Windows Server 2003 3: Windows Server 2008 4: Windows Server 2008 R2 5: Windows Server 2012 6: Windows Server 2012 R2 7: Windows Server 2016, preview **forest_level** returns the functional level of the forest that contains the bound domain.

Return Value

This command returns a forest name, server name, security identifier, or functional level depending on the provided argument.

Examples

get_bind_info acme.com server

This example returns the name of the domain controller: adserve02.acme.com

Related Commands

The following commands perform actions related to this command:

- bind binds ADEdit to a domain for subsequent ADEdit commands.
- pop restores the context from the top of ADEdit's context stack to ADEdit.
- push saves ADEdit's current context to ADEdit's context stack.
- show returns the current context of ADEdit, including its bound domains and its currently selected objects.

get_child_zones

Use the get_child_zones command to return a Tcl list of the child zones, computer roles, and computer zones for the currently selected zone stored in memory. The options to return child zones and computer roles are only applicable when you are working with hierarchical zones.

In classic zones, you can use this command to return a Tcl list of classic-computer zones under the currently selected classic zone. A classic-computer zone is a special zone type that contains a single computer to enable computer-level role assignments. The classic zone must have the corresponding computer object and that computer must be identified as a classic-computer zone to support computer-specific role assignments.

Because classic zones do not have child zones or computer roles, executing get_child_zones with the -crole or -tree option without the computer option returns an empty list.

Zone Type

Classic and hierarchical

Syntax

get_child_zones [-tree] [-crole] [-computer]

Abbreviation

gcz

Options

This command takes any of the following options:

-tree	Returns a Tcl list of the current zone's child zones. If the currently selected zone is a classic zone, this option is ignored.
-crole	Returns a Tcl list of the current zone's hosted computer roles. If the currently selected zone is a classic zone, this option is ignored.
-computer	Returns a Tcl list of the current zone's computer-specific zones. For classic zones, this option returns a list of classic-computer zones.

If you don't specify an option and the currently selected zone is a hierarchical zone, get_child_zones returns the complete list of child zones including computer roles and computerspecific "zones" that enable computer-specific overrides. If you don't specify an option and the currently selected zone is a classic zone, get_child_zones returns the list of classic-computer zones.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of child zones, computer roles, or computer-specific zones depending on the options used.

Examples

get_child_zones

This example returns:

```
{CN=cz1,CN=Zones,CN=Acme,CN=Program Data,DC=eel,DC=nest}
{CN=cz2,CN=Zones,CN=Acme,CN=Program Data,DC=eel,DC=nest}
{CN=global,CN=Zones,CN=Acme,CN=ProgramData,DC=eel,DC=nest/oracleServers}
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the zone to work with:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.
- select_zone retrieves a zone from Active Directory and stores it in memory as the currently selected zone.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates administrative rights to a specified user or group.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.

- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

get_dz_commands

Use the get_dz_commands command to check Active Directory and return a Tcl list of UNIX command objects defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use the list_dz_commands command to output to stdout.

You can only use the get_dz_commands command if the currently selected zone is a classic4 or hierarchical zones. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

get_dz_commands

Abbreviation

gdzc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of UNIX commands defined in the currently selected zone.

Examples

get_dz_commands

This example returns the list of commands: root_any

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command to work with:

- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- get_dzc_field reads a field value from the currently selected command.
- save_dz_command saves the selected command with its current settings to Active Directory.
- set_dzc_field sets a field value in the currently selected command.

get_dzc_field

Use the get_dzc_field command to return the value for a specified field from the currently selected command object that is stored in memory.

The get_dzc_field command does *not* query Active Directory for the command. If you change field values using ADEdit without saving the command to Active Directory, the field value you retrieve using get_dzc_field won't match the same field value for the command stored in Active Directory.

You can only use the get_dzc_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_dzc_field field
```

Abbreviation

gdzcf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are: **description**: Returns text describing the UNIX command. **cmd**: Returns the restricted shell command string or strings. **path**: Returns the path to the command's location. **form**: Returns an integer that indicates whether the cmd and path strings use wild cards (0) or a regular expression (1). **dzdo_runas**: Returns a list of users and groups that can run this command under dzdo version of sudo. Users may be listed by user name or user ID (UID). **dzsh_runas**: Returns a list of users and groups that can run this command in a restricted shell environment (dzsh). Users can be listed by user name or UID. You cannot get this field value if the selected zone is a classic4 zone. **keep**: Returns a comma-separated list of environment variables from the current user's environment to keep. **del**: Returns a comma-separated list of environment variables from the current user's environment to delete. **add**: Returns a comma-separated list of environment variables to add to the final set of environment variables. **pri**: Returns an integer that specifies the command priority for the restricted shell command object. **umask**: Returns an integer that defines who can execute the command. **flags**: Returns an integer that specifies a combination of different properties for the command. **createTime**: Returns the time and date this command was created, returned in generalized time format. **modifyTime**: Returns the time and date this command was last modified, returned in generalized time format. **dn**: Returns the command's distinguished name. **selinux_role**: Returns the SELinux role used when constructing a new security context for command execution (tree zone only). **selinux_type**: Returns the SELinux type used when constructing a new security context for command execution (tree zone only). **digest**: Returns the SHA-2 digest to verify the file checksum before command execution. Note that selinux_role and selinux_type are only supported on Red Hat Enterprise Linux systems and effective only on systems with SELinux enabled and joined to a hierarchical zone.

Getting the cmd and path field values

If you specify the cmd and path fields, the return value can be a string that uses wild cards (*, ?, and !), or a regular expression. If the cmd and path strings use wild cards, an asterisk (*) matches zero or more characters, a question mark (?) matches exactly one character, and the exclamation mark (!) negates matching of the specified string.

For both the cmd and path fields, the form field indicates whether the specified string is interpreted as a regular expression or as a string that includes wild cards.

Getting environment variable field values

If you specify the keep, del, or add field, the return value is a comma-separated list of environment variables. The keep, del, and add fields control the environment variables used by the commands specified by the cmd string. The keep and del settings are mutually exclusive:

- The keep field only takes effect if the flag 16 is included in the setting for the flag field.
- The del field only takes effect if the flag 16 is not included in the setting for the flag field.

Any environment variables kept or deleted are in addition to the default set of the user's environment variables that are either retained or deleted. The default set of environment variables to keep is defined in the dzdo.env_keep configuration parameter in the centrifydc.conf file. The default set of environment variables to delete is defined in the dzdo.env_delete configuration parameter in the centrifydc.conf file.

The add field returns the environment variables added to the final set of environment variables resulting from the keep or del fields.

Getting the command priority field value

If you specify the pri field, the return value indicates the command priority when there are multiple matches for command strings in a command object. If there are multiple commands specified by this command object, the pri field specifies the specifies their relative priority. The higher the value returned by this field, the higher the command's priority.

Getting the umask field value

If you specify the umask field, the return value is a 3-digit octal value that defines who can read, write, and execute the selected command object. The three digits of the umask field specify the read, write, or execute permission for the file owner, group, and other users. The left digit defines the owner execution rights, the middle digit defines the group execution rights, and the right digit defines execution rights for other users. Each digit is a combination of binary flags, one flag for each right as follows:

- 4 is read
- 2 is write
- 1 is execute

These values are added together to define the rights available for each entity. For example, a umask value of 600 indicates read and write permission (4+2) for the owner, but no permissions for the group or other users. Similarly, a umask value of 740 indicates read, write, execute permissions (4+2+1) for the owner, read permissions for the group, but no permissions for other users.

Getting command properties from the flags field value

If you specify the flags field, the return value is an integer that defines a combination of binary flags, with one flag for each of the following properties:

1—Prevents nested command execution. If this flag value is not set, nested command execution is allowed.

2—Requires authentication with the user's password.

4—Requires authentication with the run-as user's password.

8—Preserves group membership. If this flag value is not set, group membership is not preserved.

16—Resets environment variables for the command, deleting the variables specified in the dzdo.env_delete parameter and keeping the variables specified in the keep field. If this flag is not set, the command removes the unsafe environment variables specified in the dzdo.env_delete parameter along with any additional environment variables specified by the del field.

32—Requires multi-factor authentication to execute the command.

64—Prevents navigation up the path hierarchy when executing the command.

These values are added together to define the value for the flags field. For example, a flags field value of 11 indicates that nested command execution is not allowed (1), the command requires authentication using the user's password (2), and the user's group membership should be preserved (8). The value returned is the sum of these flags (1+2+8).

Return Value

This command returns a field value, which varies in type depending on the data type stored by the field.

Examples

```
get_dzc_field dzdo_runas
```

returns: root

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command to work with:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- save_dz_command saves the selected command with its current settings to Active Directory.
- set_dzc_field sets a field value in the currently selected command.

get_group_members

Use the get_group_members command to check the Active Directory group membership for a specified group. You can use this command to return a Tcl list of the users in a specified group in one of two ways:

- With the -ad option to return a simplified list of the distinguished names that are members of the specified group. The -ad option lists the users and groups that are members of the specified group without recursively expanding the group membership of any nested group.
- Without the -ad option to return a complete list of users that are members of the specified group. If you don't specify the -ad option, the command recursively expands the groups that are members of the specified group to identify all of the users in any nested group.

Zone Type

Not applicable

Syntax

```
get_group_members [-ad | -upn] group_UPN
```

Abbreviation

ggm

Options

This command takes the following options:

-ad	Returns the distinguished names for the users and groups that are members of the specified group. This option does not expand the group membership to list users who are members of nested groups.
-upn	Returns user names in user principal name (UPN) format for all of the users that are members of the specified group. This option expands the group membership of the specified group to include users who are members of nested groups. If you don't specify this option, a complete list of user names is returned using the default sAMAccount@domain format.

Arguments

This command takes the following argument:

group_UPN	string	Required. Specifies the user principal name (UPN) of the group to for which you want to return user membership.

Return Value

This command returns a Tcl list of group members.

Examples

```
get_group_members poweradmins@acme.com
```

This example returns the complete list of users who are members of the poweradmin@acme.com group, including users who are members of any nested groups, using the sAMAccountName@domain.name format. For example:

```
martin.moore@acme.com rachel.roberts@acme.com frank.smith@acme.com
```

The following example returns the distinguished names of the users and groups that are members of the demo-qa-lab@acme.com group without listing the members of any nested groups.

```
get_group_members -ad demo-qa-lab@acme.com
```

For example, this command returns the list of users and groups without expanding the group membership for the LabAdmins and QA groups:

```
CN=LabAdmins,CN=Users,DC=acme,DC=com {CN=Chris Howard,CN=Users,DC=acme,DC=com} CN=QA,CN=Users,DC=acme,DC=com CN=frank.smith,CN=UsersDC=acme,DC=com
```

Related Commands

The following commands perform actions related to this command:

- joined_get_user_membership checks Active Directory through adclient and returns a Tcl list of groups that a user belongs to.
- joined_user_in_group checks Active Directory through adclient to see if a user is in a group.
- get_effective_groups checks Active Directory and returns a Tcl list of groups a user belongs to.

get_local_group_profile_field

Use the get_local_group_profile_field command to display the value of the specified field for the currently selected local UNIX or Linux group that has a profile defined in the current zone. Before executing this command, you must select a local group by executing the select_local_group_profile command.

Zone Type

Hierarchical only.

Syntax

```
get_local_group_profile_field field_name
```

Abbreviation

```
glgpf
```

Options

This command takes no options.

Arguments

This command takes the following argument:

field_name	Required. Specifies the local group field to retrieve. The data type depends on the field. The possible values are: **gid**: The numeric group identifier. **name**: The UNIX name of the group. **member**: The UNIX name of at least one group member. **profileflag**: The value of the group's profile flag as set in the group object in the zone. For the group to be managed by the agent, the profile flag must be set to 1 or 3.If set to 1, the group profile is enabled. If the group profile is complete and the profile flag is set to 1, the profile will be installed or updated in /etc/group at the next local account refresh interval. If set to 3, the group profile is removed from /etc/group at the next local account refresh interval. **dn**: The distinguished name of the group. **createTime**: The creation time of the group profile.

modifyTime: The most recent modification time of the group profile. You can also specify AIX extended attributes as the field to get an extended attribute value for a group. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be retrieved by specifying aix.admin as the field.

Return Value

This command returns the value of the specified field.

Examples

The following example returns the GID of the currently selected local group in the zone.

```
get_local_group_profile_field gid
```

The following example returns the value of the profile flag for the currently selected local group. In this example, the profile flag is 1, meaning that the group profile in /etc/group will be updated with the latest settings from the local account zone object at the next local account refresh interval.

```
get_local_group_profile_field profileflag

1
```

If the current group is on AIX, you can get group extended attributes and values. For example, to find out if the current group is an administrative group, you can get the admin extended attribute:

```
get_local_group_profile_field aix.admin

true
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

get_local_groups_profile

Use the get_local_groups_profile command to return a TCL list of profiles for local groups that are defined in the currently selected zone.

Zone Type

Hierarchical only.

Syntax

```
get_local_groups_profile
```

Abbreviation

glgp

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

If you run this command from the command line, it returns a TCL list of profiles for local groups that are defined in the currently selected zone. The list is sorted by group UNIX name.

If you run this command in a script, no output is returned to stdout, and no output appears in the shell where the script is executed. To return output to stdout from a script, use the list_local_groups_profile command.

Examples

The following example shows a TCL list of profiles for local groups that are defined in the current zone.

```
get_local_groups_profile
lg001 lg002 lg003 lg005 lg006 lg007
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

get_local_user_profile_field

Use the get_local_user_profile_field command to display the value of the specified field for the currently selected local UNIX or Linux user that has a profile defined in the current zone. Before executing this command, you must select a local user by executing the select_local_user_profile command.

Zone Type

Hierarchical only.

Syntax

```
get_local_user_profile_field field_name
```

Abbreviation

glupf

Options

This command takes no options.

Arguments

This command takes the following argument:

field_name	string	Required. Specifies the local user profile field to retrieve. The possible values include: **uid**: The user's numeric identifier. **gid**: The GID of the user's primary group. **shell**: The local user's default shell on the local computer. Possible values are: /bin/bash, /bin/csh, /bin/ksh, /bin/sh, /bin/tcsh, %. **home**: The local user's default home directory on the local computer. **gecos**: General information about the local user account. **uname**: The UNIX name of the user. **dn**: The distinguished name of the user. **createTime**: The creation time of the user profile. **modifyTime**: The most recent modification time of the user profile. **profileflag**: The value of the user's profile flag as set in the user object in the zone. For the user to be managed by the agent, the profile flag must be set to 1 , 2, or 3. You can also specify AIX extended attributes as the field to get an extended attribute value for a user. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be retrieved by specifying aix.admin as the field.

Return Value

This command returns the value of the specified field.

Examples

The following example returns the UID of the currently selected local user in the zone.

```
get_local_user_profile_field uid
```

The following example returns the value of the profile flag for the currently selected local user. In this example, the profile flag is 2, meaning that the user profile in /etc/passwd will be updated with the latest settings from the local account zone object at the next local account refresh interval, but the password entry in /etc/passwd will be set to !! so that the user cannot log into the local computer.

```
get_local_user_profile_field profileflag
```

```
2
```

For more information about the meaning of the profile flag value, see set_local_user_profile_field.

You can also specify AIX extended attributes as the field to get an extended attribute value for a user. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be retrieved by specifying aix.admin as the field.

```
get_local_user_profile_field aix.admin
```

```
false
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.

- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

get_local_users_profile

Use the get_local_users_profile command to return a TCL list of profiles for local users that are defined in the currently selected zone.

Zone Type

Hierarchical only.

Syntax

```
get_local_users_profile
```

Abbreviation

```
glup
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

If you run this command from the command line, it returns a TCL list of profiles for local users that are defined in the currently selected zone. The list is sorted by user UNIX name.

If you run this command in a script, no output is returned to stdout, and no output appears in the shell where the script is executed. To return output to stdout from a script, use the list_local_users_profile command.

Examples

The following example shows a TCL list of profiles for local users that are defined in the current zone.

```
get_local_users_profile
db2011 db2012 lu001 lu002 lu003 lu004 lu006 lu007 lu008 lu009 lu012 lu013
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the

group object.

- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

get_nis_map

Use the get_nis_map command to return a Tcl list containing the entries for the currently selected NIS map stored in memory. This command does not return the contents of the comment field. If you want to retrieve the comment, use get_nis_map_with_comment instead.

The get_nis_map command does *not* query Active Directory for this NIS map, but changing map entries using add_map_entry and delete_map_entry changes both selected NIS map in memory and the corresponding NIS map in Active Directory so their contents should match.

Zone Type

Not applicable

Syntax

get_nis_map

Abbreviation

gnm

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of NIS map entries. Each entry contains:

- The key
- The instance number of the key (there may be multiple entries with the same key)
- The value

Each entry component is separated from the next by a colon (:).

Examples

get_nis_map

This example returns the list of map entries. For example:

{Finance:1: Hank@acme.com,jane@acme.com,joe@acme.com} {Mktg:1: Mike@acme.com,Sue@acme.com}

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and manage NIS maps:

- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout all NIS maps in the currently selected zone.

- new_nis_map creates a new NIS map and stores it in memory.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map's entries:

- add_map_entry or add_map_entry_with_comment adds an entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- get_nis_map_with_comment returns a Tcl list of the entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout of the entries in the currently selected NIS map.

get_nis_map_field

Use the get_nis_map_field command to return the value for a specified field from the currently selected NIS map stored in memory. The get_nis_map_field command does *not* query Active Directory for the NIS map. If you've changed field values using ADEdit without saving the NIS map to Active Directory, the field value you retrieve using get_nis_map_field won't match the same field value for the NIS map stored in Active Directory.

Zone Type

Not applicable

Syntax

```
get_nis_map_field field
```

Abbreviation

gnmf

Options

This command takes no options.

Arguments

This command takes the following argument, which is case-sensitive:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are: **createTime**: Specifies the time and date this NIS map was created, returned in generalized time format **modifyTime**: Specifies the time and date this NIS map was last modified, returned in generalized time format **dn**: Specifies the NIS map's distinguished name

Return Value

This command returns a field value, which varies in type depending on the data type stored by the field.

Examples

```
get_nis_map_field createTime
```

This example returns the value of the createTime field. For example: 20110525163718.0Z

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and manage NIS maps:

- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.

- list_nis_maps lists to stdout all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map's entries:

- add_map_entry or add_map_entry_with_comment adds an entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- get_nis_maps or get_nis_map_with_comment returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_map or list_nis_map_with_comment lists to stdout of the entries in the currently selected NIS map.

get_nis_map_with_comment

Use the get_nis_map command to return a Tcl list containing the entries for the currently selected NIS map stored in memory. This command includes the comment field for map entries. The get_nis_map_with_comment command does *not* query Active Directory for this NIS map, but changing map entries using add_map_entry and delete_map_entry changes both selected NIS map in memory and the corresponding NIS map in Active Directory so their contents should match.

Zone Type

Not applicable

Syntax

get_nis_map_with_command``

Abbreviation

gnmwc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of NIS map entries. Each entry contains:

- The key
- The instance number of the key (there may be multiple entries with the same key)
- The value
- The comment

Each entry component is separated from the next by a colon (:).

Examples

get_nis_map_with_comment

This example returns the map entries including comments:

{Finance:1: Hank@acme.com,jane@acme.com,joe@acme.com: Finance dept staff}{Mktg:1: Mike@acme.com,Sue@acme.com: Marketing dept staff}

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and manage NIS maps:

- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map's entries:

- add_map_entry or add_map_entry_with_comment adds an entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_map or list_nis_map_with_comment lists to stdout of the entries in the currently selected NIS map.

get_nis_maps

Use the get_nis_maps command to check Active Directory and return a Tcl list of NIS maps defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_nis_maps to output the list of NIS maps to stdout.

Zone Type

Not applicable

Syntax

get_nis_maps

Abbreviation

gnms

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of NIS maps defined in the currently selected zone.

Examples

get_nis_maps

This example returns the list of NIS maps: Aliases Printers Services

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage NIS maps:

- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- list_nis_maps lists to stdout all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the other commands to work with that map's entries.

get_object_field

Use the get_object_field command to return the value of a specified field from the currently selected Active Directory object stored in memory. The get_object_field command does *not* query Active Directory for the object. If you change field values using ADEdit without saving the object to Active Directory, the field value you retrieve using get_object_field won't match the same field value for the object stored in Active Directory.

Zone Type

Not applicable

Syntax

get_object_field field

Abbreviation

gof

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values include any attribute that can be defined for the type of object currently selected. Special values are: **sid**: The object's security identifier. **guid**: The object's globally unique identifier. **sd**: The object's security descriptor. **createTime**: The time and date this object was created, returned in generalized time format. **modifyTime**: The time and date this object was last modified, returned in generalized time format. **dn**: The object's distinguished name.

Return Value

This command returns a field value, which varies in type depending on the data type stored by the field.

Examples

get_object_field guid

This example returns the globally unique identifier for an object. For example:

44918ee7-80bc-4741-95d3-dd189e235ab8

Related Commands

Before you use this command, you must have a currently selected Active Directory object stored in memory. The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of matching objects.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object with its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.

- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field_names returns a Tcl list of the field names (attributes) for the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_field sets a field value in the currently selected Active Directory object.

get_object_field_names

Use the get_object_field_names command to return a Tcl list of the field names for each of the fields—the object attributes—of the currently selected Active Directory object. The get_object_field_names command does not query Active Directory for the object's field names but looks at the selected object as it is stored in ADEdit memory.

Zone Type

Not applicable

Syntax

get_object_field_names

Abbreviation

gofn

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of field names.

Examples

select_object "cn=amy adams,cn=users,dc=ajax,dc=com" get_object_field_names

This example returns the field names associated with the selected user Amy Adams:

_SID _dn _objectCategory _server accountExpires cn codePage countryCode distinguishedName
gidNumber instanceType lastLogonTimestamp loginShell msDS-MembersForAzRoleBL msSFU30NisDomain
nTSecurityDescriptor name objectCategory objectClass objectGUID objectSid primaryGroupID
pwdLastSet sAMAccountName sAMAccountType uSNChanged uSNCreated uid uidNumber unixHomeDirectory
userAccountControl userPrincipalName whenChanged whenCreated

Related Commands

Before you use this command, you must have a currently selected Active Directory object stored in memory. The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects that match the search criteria.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object and its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.

- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_field sets a field value in the currently selected Active Directory object.

get_objects

Use the get_objects command to perform an LDAP search of Active Directory and return a Tcl list of the distinguished names (DNs) of the objects that match the search criteria. You specify a container in Active Directory where the search begins and a standard LDAP filter that defines the objects you're searching for.

You can control the nature of the search through options that specify whether to use the global catalog (GC) for a forest-wide search, the number of levels deep for the search to go below the beginning container of the search, and the maximum number of objects for the get_objects command to return.

Zone Type

Not applicable

Syntax

get_objects [-gc] [-depth one|sub] [-limit limit] [-f forest] base filter

Abbreviation

go

Options

This command takes the following options:

-gc	Requests a forest-wide search using a global catalog. For this option to work, ADEdit must be bound to a global catalog domain controller using the bind command with the -gc option. If you don't specify this option, the search is only within the currently bound domains.
-depth one | sub	Specifies how deep to search. This option must be followed by one of two values: **one**: Specifies that the search will search only through objects immediately below the container specified by the argument *base*. **sub**: Specifies that the search will be full-depth, starting at the container specified by *base* and continuing through all sub-containers below that level. If you don't specify this option, the search defaults to the value **one**.
-limit *limit*	Limits the number of objects returned by the search to the positive integer specified by *limit*. If you don't specify this option, the search returns all matching objects without limit.
-f *forest*	Specifies the forest to search. If you bind ADEdit to multiple forests, you can use this option to identify a specific forest to search for objects matching the criteria you specify.

Arguments

This command takes the following arguments:

base	DN	Required. Specifies the distinguished name of an Active Directory container in which to start the search. If you want to perform a forest-wide search using the global catalog option but do not specify the forest to search, use an empty string as the base argument. For example: get_objects -gc -depth sub "" (cn=demo). You should not use an empty string as the starting point for a search if you bind to multiple forests. If you bind to multiple forests, you should always specify the forest to search.
filter	LDAP filter	Required. A string that uses standard LDAP filter syntax to specify criteria for the search.

Return Value

This command returns a Tcl list of the distinguished names of the objects matching the search criteria.

Examples

```
get_objects “cn=users,dc=acme,dc=com” (objectclass=*)
```

This example returns a list of distinguished name matching the objectclass filter:

```
CN=Builtin,DC=acme,DC=com CN=Computers,DC=acme,DC=com {OU=Domain Controllers,DC=acme,DC=com} CN=ForeignSecurityPrincipals,DC=acme,DC=com
CN=Infrastructure,DC=acme,DC=com CN=LostAndFound,DC=acme,DC=com {CN=NTDS Quotas,DC=acme,DC=com} {CN=Program Data,DC=acme,DC=com} CN=System,DC=acme,DC=com
CN=Users,DC=acme,DC=com
```

Related Commands

The following commands enable you to view and select the object to work with:

- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object and its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_field sets a field value in the currently selected Active Directory object.

get_pam_apps

Use the get_pam_apps command to check Active Directory and return a Tcl list of plug-in authentication module (PAM) applications defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_pam_apps to output the list of PAM applications to stdout.

You can only use the get_pam_apps command to return information about PAM applications if the currently selected zone is a classic4 or hierarchical zones. The command does not work for other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_pam_apps
```

Abbreviation

```
gpam
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of PAM applications defined in the currently selected zone. Each element in the string is the name of a PAM application.

Examples

```
get_pam_apps
```

This example returns all of the PAM application rights for the selected zone:

```
dzssh-all dzssh-direct-tcpip dzssh-exec dzssh-scp dzssh-sftp dzssh-shell dzssh-subsystem dzssh-tcpip-forward dzssh-tunnel dzssh-x11-forwarding loginall ssh sshd
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. After you have a zone stored in memory, you can use the following commands to view and select the PAM application to work with:

- list_pam_apps lists to stdout the PAM application rights in the current zone.
- new_pam_app creates a new PAM application right and stores it in memory.
- select_pam_app retrieves a PAM application from Active Directory and stores it in memory.

After you have a PAM application stored in memory, you can use the following commands to work with that PAM application's attributes, delete the PAM application, or save information for the PAM application:

- delete_pam_app deletes the selected PAM application from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application.
- save_pam_app saves the selected PAM application with its current settings to Active Directory.
- set_pam_field sets a field value in the currently selected PAM application.

get_pam_field

Use the get_pam_field command to return the value of a specified field for the currently selected plug-in authentication module (PAM) application object stored in memory. The get_pam_field command does *not* query Active Directory for the PAM application. If you change field values using ADEdit without saving the PAM application to Active Directory, the field value you retrieve using get_pam_field won't match the same field value for the PAM application stored in Active Directory.

You can only use the get_pam_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_pam_field field
```

Abbreviation

```
gpf
```

Options

This command takes no options.

Arguments

This command takes the following argument:

	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are: **application**: The name of

field	string	the application allowed to use adclient's PAM authentication service. The name can be literal, or it can contain ? or * wildcard characters to specify multiple applications. **description**: Text describing the PAM application. **createTime**: The time and date this PAM application was created, returned in generalized time format. **modifyTime**: The time and date this PAM application was last modified, returned in generalized time format. **dn**: the PAM application's distinguished name.

Return Value

This command returns a field value. The data type for this value depends on the field specified.

Examples

```
get_pam_field application
```

This example returns the contents of the application field:

```
ftp
```

The selected PAM application object specifies ftp can authenticate using adclient.

Related Commands

Before you use this command, you must have a currently selected PAM application object stored in memory. The following commands to view and select the PAM application to work with:

- get_pam_apps returns a Tcl list of PAM application rights in the current zone.
- list_pam_apps lists to stdout the PAM application rights in the current zone.
- new_pam_app creates a new PAM application right and stores it in memory.
- select_pam_app retrieves a PAM application right from Active Directory and stores it in memory.

After you have a PAM application stored in memory, you can use the following commands to work with that PAM application's attributes, delete the PAM application, or save information for the PAM application:

- delete_pam_app deletes the selected PAM application right from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application right.
- save_pam_app saves the selected PAM application right with its current settings to Active Directory.
- set_pam_field sets a field value in the currently selected PAM application right.

get_parent_dn

Use the get_parent_dn command to specify an LDAP path using a distinguished name (DN) and return the parent of the path. This command removes the first element from the distinguished name and returns the rest of the DN.

Zone Type

Not applicable

Syntax

```
get_parent_dn DN
```

Abbreviation

```
gpd
```

Options

This command takes no options.

Arguments

This command takes the following argument:

DN	string	Required. Specifies a distinguished name.

Return Value

This command returns a distinguished name that is the parent of the supplied distinguished name.

Examples

```
get_parent_dn CN=global,CN=Zones,CN=Acme,DC=acme,DC=com
```

This example returns: `CN=Zones,CN=Acme,DC=acme,DC=com`

Related Commands

The following command performs actions related to this command:

- `get_rdn` returns the relative distinguished name of a specified LDAP path.

get_pending_zone_groups

Use the `get_pending_zone_groups` command to check Active Directory and return a Tcl list of pending import groups for the currently selected zone. Pending import groups are group profiles that have been imported from Linux or UNIX computers, but not yet mapped to any Active Directory group. If executed in a script, this command does not output its list to `stdout`, and no output appears in the shell where the script is executed. Use `list_pending_zone_groups` to output the list to `stdout`.

Zone Type

Classic and hierarchical

Syntax

```
get_pending_zone_groups
```

Abbreviation

gpzg

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of pending import group profiles that have been imported into the currently selected zone. Each entry in the list contains the following fields, separated by colons : :

- Distinguished name (DN) of the pending import group as it is stored in Active Directory. The distinguished name for each pending import group includes a prefix that consists of “PendingGroup” and the globally unique identifier (GUID) for the group.
- UNIX group name.
- Numeric group identifier (GID).

Examples

```
get_pending_zone_groups
```

The command returns output in the form of:

```
DN:group_name:gid
```

This sample command might return output similar to the following:

```
CN=PendingGroup_573135e7-edd9-46b9-9cbd-c839570a90c8,CN=Groups, CN=bean_pz,CN=Zones,CN=Acme,DC=win2k3,DC=test:root:0
CN=PendingGroup_7878065a-4d2f-4749-8f3b-6ffe24303f6a,CN=Groups, CN=bean_pz,CN=Zones,CN=Acme,DC=win2k3,DC=test:unixgrp:5000
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following command performs actions related to this command:

- `select_object` retrieves the specified Active Directory object and its attributes from Active Directory and stores the object in memory.
- `get_object_field` enables you to view and work with the pending import group.

get_pending_zone_users

Use the `get_pending_zone_users` command to check Active Directory and return a Tcl list of pending import users for the currently selected zone. Pending import users are user profiles that have been imported from Linux or UNIX computers, but not yet mapped to any Active Directory user. If executed in a script, this command does not output its list to `stdout`, and no output appears in the shell where the script is executed. Use `list_pending_zone_users` to output the list to `stdout`.

Zone Type

Classic and hierarchical

Syntax

```
get_pending_zone_users
```

Abbreviation

```
gpzu
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of pending import user profiles that have been imported into the currently selected zone. Each entry in the list contains the following fields, separated by colons : :

- Distinguished name (DN) of the pending import user as it is stored in Active Directory. The distinguished name for each pending import user includes a prefix that consists of “PendingUser” and the globally unique identifier (GUID) for the user.
- UNIX user name.
- Numeric user identifier (UID).
- Numeric primary group identifier (GID).
- Personal information from the GECOS field.
- Home directory.
- Default login shell.

Examples

```
get_pending_zone_users
```

This sample command might return output similar to the following:

CN=PendingUser_09024f3a-6abc-4666-a127-722f9fe0e0bf,CN=Users,CN=finance, CN=Zones,CN=Acme,DC=win2k3,DC=test:root:0:0:root:/root:/bin/bash
CN=PendingUser_0b9fe038-1325-438f-8529-cb190ab5914a,CN=Users,CN=finance, CN=Zones,CN=Acme,DC=win2k3,DC=test:bean:6001:5000:bean.zhang:/home/bean:/bin/bash

Before you use this command, you must have a currently selected zone stored in memory. The following command performs actions related to this command:

- select_object retrieves the specified Active Directory object and its attributes from Active Directory and stores the object in memory.
- get_object_field enables you to view and work with the pending import group.

get_pwnam

Use the get_pwnam command to look up a UNIX user name in the /etc/passwd file on the ADEdit host computer. If there's an entry for the specified user name, the command returns the profile values of that entry as a Tcl list. The get_pwnam command uses the NSS layer to perform the lookup operation. You can use the command to look up information for any user in the /etc/passwd file, including root.

Zone Type

Not applicable

Syntax

get_pwnam unix_name

Abbreviation

gpn

Options

This command takes no options.

Arguments

This command takes the following argument:

unix_name	string	Required. Specifies the UNIX user name to search for in the /etc/passwd file.

Return Value

This command returns a Tcl list of user profile attributes for a specified user if the specified user name is found in the local /etc/passwd file. If the command doesn't find the specified user, it a "User not found" message.

Examples

get_pwnam adam

This example returns the profile for the UNIX user adam:

adam x 500 500 {Adam Andrews} /home/adam /bin/bash

Related Commands

The following command performs actions related to this command:

- getent_passwd returns a Tcl list of all entries in the local /etc/passwd file.

get_rdn

Use the get_rdn command to specify an LDAP path using a distinguished name (DN) and return the relative distinguished name. This command returns only the first element of the supplied distinguished name.

Zone Type

Not applicable

Syntax

```
get_rdn DN
```

Abbreviation

```
grdn
```

Options

This command takes no options.

Arguments

This command takes the following argument:

DN	string	Required. Specifies a distinguished name.

Return Value

This command returns the first element of the supplied distinguished name.

Examples

```
get_rdn CN=global,CN=Zones,CN=Acme,DC=acme,DC=com
```

This example returns: `CN=global`

Related Commands

The following command performs actions related to this command:

- `get_parent_dn` returns the parent distinguished name of a specified LDAP path.

get_role_apps

Use the `get_role_apps` command to return a Tcl list of PAM application rights associated with the currently selected role.

The `get_role_apps` command does *not* query Active Directory for the role. If you change the PAM applications associated with the current role using ADEdit without saving the role to Active Directory, the PAM applications you retrieve using `get_role_apps` won't match the same PAM applications for the role as stored in Active Directory.

You can only use the `get_role_apps` command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_role_apps
```

Abbreviation

grap

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of PAM applications associated with the currently selected role. Each PAM application in the list shows the application name followed by a slash (/) and the zone in which the PAM application is defined.

Examples

```
get_role_apps
```

This example returns the list of PAM applications for the currently selected role: ftp/cz1

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands to view and select the role to work with:

- get_roles returns a Tcl list of roles in the currently selected zone.
- list_roles lists to stdout the roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role's attributes, delete the role, or save information for the role:

- add_command_to_role adds a UNIX command to the currently selected role.
- add_pamapp_to_role adds a PAM application to the currently selected role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_commands returns a Tcl list of the UNIX commands associated with the currently selected role.
- get_role_field reads a field value from the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the currently selected role.
- remove_command_from_role removes a UNIX command from the currently selected role.
- remove_pamapp_from_role removes a PAM application from the currently selected role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the currently selected role.

get_role_assignment_field

Use the get_role_assignment_field command to return the value for a specified field from the currently selected role assignment stored in memory. The get_role_assignment_field command does *not* query Active Directory for the role assignment. If you change field values using ADEdit without saving the role assignment to Active Directory, the field value you retrieve using get_role_assignment_field won't match the same field value for the role assignment stored in Active Directory.

You can only use the get_role_assignment_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_role_assignment_field field
```

Abbreviation

graf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are:
		assignee: Returns user display name in format specific to type of logged in user.
		customAttr: Returns the custom text strings set for the role assignment.
		customAttr: Returns the custom text strings set for the role assignment.
		description: Returns the description for the role assignment.
		dn: Returns the role assignment's distinguished name.
		from: Returns the starting date and time for the role assignment. The start and end dates and times are expressed in standard UNIX time. You can use the Tcl clock command to manipulate these values. A value of 0 indicates no date or time is set for the role assignment.
		modifyTime: Returns the time and date this role assignment was last modified, returned in generalized time format.
		ptype: Returns a letter or symbol that indicates the account type associated with a role assignment. You can use the explain_ptype command to translate the returned value into a text string that describes the account type.
		role: Returns the name of the role and the zone in which the role is defined.
		to: Returns the ending date and time for the role assignment.

Return Value

This command returns a field value. The data type depends on the field specified.

Examples

This example returns the role name (root) and the zone where the role is defined (global):

```
get_role_assignment_field role
```

```
root/global
```

This example returns the assignee display name in the appropriate format.

```
get_role_assignment_field assignee
```

- For AD user/group:

  ```
  CN=dc1,CN=Users,DC=sayms,DC=local
  ```

- For trusted forest AD user/group:

  ```
  CN=S-1-5-21-4259971489-770964042-439865176-1106,CN=ForeignSecurityPrincipals,DC=sayms,DC=local
  ```

- For local uid:

  #56789@localhost

- For local user:

  localuser1@localhost

- For local group:

  %localgroup1@localhost

Related Commands

Before you use this command, you must have a currently selected role assignment stored in memory. The following commands to view and select the role assignment to work with:

get_role_assignments

Use the get_role_assignments command to check Active Directory and return a Tcl list of role assignments defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_role_assignments to output the list to stdout.

If you do not specify an option, the command returns the current users and groups in the zone with a role assignment.

You can only use the get_role_assignments command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

get_role_assignments [-upn] [-user] [-group] [-invalid]

Abbreviation

gra

Options

This command takes any one of the following options:

-upn	Returns user names in user principal name (UPN) format, not the default sAMAccount@domain format.
-user	Returns a Tcl list of the current users in the zone with a role assignment.
-group	Returns a Tcl list of the current groups in the zone with a role assignment.
-invalid	Returns a Tcl list of any invalid role assignments in the zone. For example, this option would return role assignment for a group or user that no longer exists.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of role assignments defined in the currently selected zone. Each role assignment includes the sAMAccount@domain name or the

user principal name of the user or group to whom the role is assigned, the name of the role assigned, and the zone in which the role is defined. These three pieces of data are separated from each other by a slash (/).

Examples

```
get_role_assignments
```

This example returns the list of role assignments:

```
poweradmins@acme.com/root/global proj_admins@acme.com/login/global
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. After you have a zone stored in memory, you can use the following commands to view and select the role assignment to work with:

- list_role_assignments lists to stdout the role assignments in the current zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment's attributes, delete the role assignment, or save information for the role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the currently selected role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- set_role_assignment_field sets a field value in the currently selected role assignment.
- write_role_assignment saves the selected role assignment to a file.

get_role_commands

Use the get_role_commands command to return a Tcl list of UNIX commands associated with the currently selected role. The get_role_commands command does *not* query Active Directory for the role. If you change commands associated with the current role using ADEdit without saving the role to Active Directory, the commands you retrieve using get_role_commands won't match the same commands for the role stored in Active Directory.

You can only use the get_role_commands command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
get_role_commands
```

Abbreviation

```
grc
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of commands associated with the currently selected role. Each command in the list shows the command name followed by a slash (/) and the zone in which the command is defined.

Examples

get_role_commands

This example returns the list of commands:

pwd/global ls/global cd/childzone1

Related commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role's attributes, delete the role, or save information for the role:

- add_command_to_role adds a UNIX command to the currently selected role.
- add_pamapp_to_role adds a PAM application to the currently selected role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications associated with the currently selected role.
- get_role_field reads a field value from the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the currently selected role.
- remove_command_from_role removes a UNIX command from the currently selected role.
- remove_pamapp_from_role removes a PAM application from the currently selected role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the currently selected role.

get_role_field

Use the get_role_field command to return the value for a specified field from the currently selected role stored in memory. The get_role_field command does *not* query Active Directory for the role. If you change field values using ADEdit without saving the role to Active Directory, the field value you retrieve using get_role_field won't match the same field value for the role stored in Active Directory.

You can only use the get_role_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

get_role_field field

Abbreviation

grf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve.

The possible field values are:

- **allowLocalUser**: Returns true or false depending on whether local users can be assigned to the role. You cannot get this field value if the selected zone is a classic4 zone.
- **AlwaysPermitLogin**: Returns true or false depending on whether "rescue rights" are configured for the role. You cannot get this field value if the selected zone is a classic zone.
- **auditLevel**: Returns the auditing level configured for the role. Roles can be configured without auditing (not requested), to audit if possible, or to have auditing required. You cannot get this field value if the selected zone is a classic4 zone.
- **createTime**: Returns the time and date this role was created in generalized time format.
- **customAttr**: Returns the custom text strings set for the role.
- **description**: Returns the text string that describes the role.
- **dn**: Returns the role's distinguished name.
- **modifyTime**: Returns the time and date this role was last modified in generalized time format.
- **sysrights**: Returns the system rights granted to the role. This value is an integer that represents a combination of binary flags, one for each system right. You cannot get this field value if the selected zone is a classic zone.

  For more information about the value returned for system rights, see the section below, Getting the System Rights Field for a Role.
- **timebox**: Returns the hours and days in the week when the role is enabled. This value is a 42-digit hexadecimal number.

  When represented in binary, each bit represents an hour of the week as described in the Timebox Value Format
- **visible**: Returns true or false depending on whether "User is visible" right is configured for the role. You cannot get this field value if the selected zone is a classic zone.

Getting the system rights field for a role

You can specify the sysrights field to return information about the system rights that have been granted to the currently selected role. This field value is an integer that represents a combination of binary flags, with one flag for each of the following system rights:

1—Password login and non password (SSO) login are allowed.

2—Non password (SSO) login is allowed.

4—Account disabled in Active Directory can be used by sudo, cron, etc.

8—Log in with non-restricted shell.

16—Audit not requested/required.

32—Audit required.

64—Always permit to login.

128—Remote login access is allowed for Windows computers.

256—Console login access is allowed for Windows computers.

512—Require multi-factor authentication through the Delinea Connector to log on.

1024—PowerShell remote access is allowed

These values are added together to define the sysrights field value. For example, a sysrights value of 6 indicates that the role is configured to allow single sign-on

login and to ignore disabled accounts (2+4). A value of 11 indicates that the most common UNIX system rights are enabled (1+2+8). A value of 384 indicates that most common Windows system rights are enabled (128+256).

Return Value

This command returns a field value, which varies in type depending on the data type stored by the field.

Examples

```
get_role_field timebox
```

This example returns the content of the timebox field:

```
00FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
```

This return value indicates that the role is enabled during all hours of the weekdays, but none of the weekends.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role's attributes, delete the role, or save information for the role:

- add_command_to_role adds a UNIX command to the currently selected role.
- add_pamapp_to_role adds a PAM application to the currently selected role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications associated with the currently selected role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the currently selected role.
- remove_command_from_role removes a UNIX command from the currently selected role.
- remove_pamapp_from_role removes a PAM application from the currently selected role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the currently selected role.

get_role_rs_commands

Use the get_role_rs_commands command to return a Tcl list of the restricted shell commands associated with the currently selected role.

The get_role_rs_commands command does not query Active Directory for the restricted shell commands. If you change the restricted shell commands associated with the current role using ADEdit without saving the role to Active Directory, the commands you retrieve using get_role_rs_commands won't match the restricted shell commands that are stored in Active Directory.

You can only use get_role_rs_commands if the currently selected zone is a classic4 zone. This command does not work in other types of zones.

Zone Type

Classic only

Syntax

```
get_role_rs_commands
```

Abbreviation

```
grrsc
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of restricted shell commands associated with the currently selected role. Each restricted shell command in the list shows the restricted shell command name followed by a slash (/) and the zone in which the restricted shell command is defined.

Examples

```
get_role_rs_commands
```

This example returns : rse1-id2/c123 rse1-id1/c123

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with restricted shells:

- get_role_rs_env returns the restricted shell environment from the currently selected role.

get_role_rs_env

Use the get_role_rs_env command to return the restricted shell environment from the currently selected role that is stored in memory.

The get_role_rs_env command does not query the data stored in Active Directory for the role. If you change the restricted shell environment in ADEdit without saving the role to Active Directory, the value you retrieve using get_role_rs_env won't match the same value for the role that is stored in Active Directory.

You can only use the get_role_rs_env command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

```
get_role_rs_env
```

Abbreviation

```
grrse
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns the restricted shell environment of the currently selected role if it runs successfully. If the currently selected role does not require a restricted shell environment, the command returns nothing.

Examples

get_role_rs_env

This example returns the restricted shell environment if it exists for the selected role:

rse1

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with restricted shells:

- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- save_rs_env saves the restricted shell environment to Active Directory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

get_roles

Use the get_roles command to check Active Directory and return a Tcl list of roles defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_roles to output the list to stdout.

You can only use the get_roles command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

get_roles

Abbreviation

getr

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of roles defined in the currently selected zone.

Examples

get_roles

This example returns the list of roles:

```
{Rescue - always permit login} scp sftp listed {UNIX Login} {Windows Login} winscp
```

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- list_roles lists to stdout the roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with role:

- add_command_to_role adds a UNIX command to the currently selected role.
- add_pamapp_to_role adds a PAM application to the currently selected role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications associated with the currently selected role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the currently selected role.
- remove_command_from_role removes a UNIX command from the currently selected role.
- remove_pamapp_from_role removes a PAM application from the currently selected role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the currently selected role.

get_rs_commands

Use the get_rs_commands command to return a Tcl list of restricted shell commands that are defined for the currently selected zone. If you want to return a list of restricted shell commands to stdout, use the list_rs_commands command.

Zone Type

Classic only

Syntax

```
get_rs_commands
```

Abbreviation

```
grsc
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of restricted shell commands for the currently selected zone.

Examples

```
get_rs_commands
```

This example returns output similar to this:

```
rse1-id1 rse1-id2 rse2-id1
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

get_rs_envs

Use the get_rs_envs command to check Active Directory and return a list of restricted environments that are defined within the currently selected zone. If you want to return a list of restricted shell environment to stdout, use the list_rs_envs command.

Zone Type

Classic only

Syntax

```
get_rs_envs
```

Abbreviation

```
grse
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of restricted environments in the currently selected zone.

Examples

```
get_rs_envs
rse1 rse2
```

This example returns the list of restricted shell environments.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

get_rsc_field

Use the get_rsc_field command to return the value of a specified field value from the currently selected restricted shell command that is stored in memory. Delinea-specific fields are similar to Active Directory attributes but are stored within the Active Directory schema.

The get_rsc_field command does not query Active Directory for the restricted shell command. If you change field values using ADEdit without saving the restricted shell command to Active Directory, the field value you retrieve using get_rsc_field won't match the value stored in Active Directory.

You can only use the get_rsc_field command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

get_rsc_field field

Abbreviation

grscf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the name of the field whose value you want to retrieve. The possible values are: **description**: Returns text describing the restricted shell command. **cmd**: Returns the restricted shell command string or strings. **path**: Returns the path to the command's location. **form**: Returns an integer that indicates whether the cmd and path strings use wild cards (0) or a regular expression (1). **dzsh_runas**: Returns a list of users and groups that can run this command in a restricted shell environment (dzsh). Users can be listed by user name or UID. **keep**: Returns a comma-separated list of environment variables from the current user's environment to keep. **del**: Returns a comma-separated list of environment variables from the current user's environment to delete. **add**: Returns a comma-separated list of environment variables to add to the final set of environment variables. **pri**: Returns a n integer that specifies the command priority for the restricted shell command object. **umask**: Returns an integer that defines who can execute the command. **flags**: Returns an integer that specifies a combination of different properties for the command. **createTime**: The time and date this command was created, returned in generalized time format. **modifyTime**: The time and date this command was last modified, returned in generalized time format. **dn**: The command's distinguished name.

Return Value

This command returns a field value. The data type depends on the field specified. For more information about the field values returned by different fields, see get_dzc_field.

Examples

get_rsc_field description

This example returns the contents of the description field:

This is the RSC description

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

get_rse_cmds

Use the get_rse_cmds command to return a Tcl list of restricted shell commands associated with the currently selected restricted shell environment.

The get_rse_cmds command does not query Active Directory for the restricted shell environment. If you change the restricted shell commands associated with the current restricted shell environment using ADEdit without saving the restricted shell environment to Active Directory, the commands you retrieve using get_rse_cmds won't match those stored in Active Directory.

You can only use the get_rse_cmds command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

get_rse_cmds

Abbreviation

grsec

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of restricted shell commands associated with the currently selected restricted shell environment. Each restricted shell command in the list shows the command name followed by a slash (/) and the zone in which the command is defined.

Examples

get_rse_cmds

The command returns the list restricted commands:

rse1-id2/c123 rse1-id1/c123

Related Commands

Before you use this command, you must have a currently selected restricted shell environment stored in memory. The following commands enable you to view and select the restricted shell environments:

- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- save_rs_env saves the restricted shell environment to Active Directory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following command to work with its fields:

- set_rse_field sets a field value in the current restricted shell environment.

get_rse_field

Use the get_rse_field command to return a field value from the currently selected restricted shell environment stored in memory.

The get_rse_field command does not query Active Directory for the restricted shell environment. If you have changed field values using ADEdit without saving the restricted shell environment to Active Directory, the field value you retrieve using get_rse_field won't match the field value for the restricted shell environment that is stored in Active Directory.

You can only use the get_rse_field command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

```
get_rse_field field
```

Abbreviation

```
grsef
```

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the name of the field whose value to get. The only possible value is: description: Returns a text string describing the restricted shell environment.

Return Value

This command returns a field value, which varies in type depending on the data type stored by the field.

Examples

```
get_rse_field description
```

This command returns the content of the description field. For example:

```
This is the restricted shell environment description
```

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to

work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

get_schema_guid

Use the get_schema_guid command to look up a specified class or attribute in Active Directory. If the specified object is found, the command returns the globally unique identifier (GUID) of the class or attribute.

This command is useful for setting a security descriptor (SD) at a class or attribute level.

Zone Type

Not applicable

Syntax

```
get_schema_guid schema_name
```

Abbreviation

gsg

Options

This command takes no options.

Arguments

This command takes the following argument:

schema_name	string	Required. Specifies the name of a class or attribute.

Return Value

This command returns the globally unique identifier (GUID) of the provided schema object (class or attribute).

Examples

```
get_schema_guid MS-DS-Az-Role
```

This example returns the globally unique identifier of MS-DS-Az-Role:

```
8213eac9-9d55-44dc-925c-e9a52b927644
```

Related Commands

None.

get_zone_computer_field

Use the get_zone_computer_field command to return the value of a specified field from the currently selected zone computer stored in memory. The get_zone_computer_field command does *not* query Active Directory for the zone computer. If you change field values using ADEdit without saving the zone computer to Active Directory, the field value you retrieve using get_zone_computer_field won't match the same field value for the zone computer stored in Active Directory.

Zone Type

Classic and hierarchical

Syntax

```
get_zone_computer_field field
```

Abbreviation

gzcf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are: **addn**: Returns the distinguished name of the Active Directory computer object for the zone computer. For example, if the computer object is created in the default Computers container, this field might return a path similar to CN=firefly-sf,CN=Computers,DC=ajax,DC=org. **agentVersion**: Returns the version of agent currently installed on the zone computer. **cpus**: Returns the number of CPUs in the computer. **createTime**: Returns the time and date this zone computer was created (in generalized time format). **dn**: Returns the distinguished name of the service connection point for the zone computer. If the computer is in a Services for UNIX (SFU) zone, no value is returned for this field. **dnsname**: Returns the domain name service (DNS) name of the zone computer. **enabled**: Returns 1 if the zone computer is enabled in its zone or 0 if it is not. **modifyTime**: Returns the time and date this zone computer was last modified (in generalized time format).

Return Value

This command returns a field value. The data type depends on the field specified.

Examples

```
get_zone_computer_field dnsname
```

This example returns the name of the zone computer as listed in DNS:

```
printserver.acme.com
```

Related Commands

Before you use this command, you must have a currently selected zone computer stored in memory. The following commands enable you to view and manage the zone computers:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

get_zone_computers

Use the get_zone_computers command to check Active Directory and return a Tcl list of zone computers defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_zone_computers to output the list to stdout.

Zone Type

Classic and hierarchical

Syntax

get_zone_computers

Abbreviation

gzc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of zone computers defined in the currently selected zone. Each entry in the list is the security identifier (SID) of a computer that you can use to look up that computer.

Examples

get_zone_computers

This example returns the security identifier for each computer:

*S-1-5-21-2076040321-3326545908-468068287-1107

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the zone computers:

- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

get_zone_field

Use the get_zone_field command to return the value for a specified field from the currently selected zone stored in memory. The get_zone_field command does *not*

query Active Directory for this zone. If you change field values using ADEdit without saving the zone to Active Directory, the field value you retrieve using get_zone_field won't match the same field value for the zone stored in Active Directory.

Zone Type

Classic and hierarchical

Syntax

get_zone_field field

Abbreviation

gzf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve.

The data type depends on the field you return. The possible field values are:

- **availableshells**: Returns the shells available to assign to new users in the zone.
- **block.parent.zgroup**: Returns the value of the block.parent.zgroup field in the zone object's description.
- **cloudurl**: Returns the name of the cloud instance associated with the selected zone.
- **computers**: Returns the computer group UPN that is assigned to the computer role selected as a zone.
- **createTime**: Returns the time and date this zone was created.
- **customAttr**: Returns the custom text strings that have been set for the zone. This field is only applicable for hierarchical zones.
- **defaultgid**: Returns the default primary group to assign to new users.
- **defaultgecos**: Returns the default GECOS data to assign to new users.
- **defaulthome**: Returns the default home directory to assign to new users.
- **defaultshell**: Returns the default shell to assign to new users.
- **description**: Returns the description of the zone.
- **dn**: Returns the zone's distinguished name.
- **gidnext**: Returns the next GID to use when auto-assigning GID numbers to new groups.
- **gidreserved**: Returns the GID number or range of numbers (1-100) that are reserved.
- **groupname**: Returns the default group name used for new groups in the zone.
- **modifyTime**: Returns the time and date this zone was last modified.
- **nisdomain**: Returns the name of the NIS domain if it has been set.
- **parent**: Returns the distinguished name (DN) of the parent zone for the selected zone.
- **schema**: Returns the schema used in this zone, for example, std.
- **sid2iddomainmap**: Returns the domain ID mapping from the selected zone. This field is not supported for auto zones nor classic zones.
- **sfudomain**: Returns the Windows domain name for the SFU zone. Only use this argument if the current zone is a Service for UNIX (sfu) zone.
- **tenantid**: Returns the Delinea Platform tenant ID for the zone. This field is only applicable for hierarchical zones.
- **type**: Returns the type of the zone, for example, classic4 or tree.
- **uidnext**: Returns the next UID to use when auto-assigning UID numbers to new users.
- **uidreserved**: Returns the UID number or range of numbers (1-100) that are reserved.
- **username**: Returns the default user name used for new users in the zone.

For more information about the values returned by these fields, see the Return value section.

Return Value

This command returns the current value for the specified field. The data type depends on the field specified.

availableshells	Returns the list of shells available to choose from when adding new users to the currently selected zone. The value is a list of shell paths, separated by colons :. For example, "/bin/bash:/bin/csh:/bin/ksh"
block.parent.zgroup	Returns the value of the block.parent.zgroup field from the zone object's description for the currently selected zone. This field can be true if you want to prevent groups provisioned in the parent zone from being visible in the child zone if they aren't being used. The default value is false.
cloudurl	Returns the fully-qualified URL of the cloud instance associated with the selected zone.
computers	Returns the computer group UPN that is assigned to the computer role if the currently selected zone is a "computer role" zone.
createTime	Returns the time and date this zone was created (in generalized time format).
defaultgid	Returns the default primary group to assign to new users in the currently selected zone. The value can be a specific GID value or include variables.
defaultgecos	Returns the default GECOS data to assign to new users in the currently selected zone. The value can be a string or include variables.
defaulthome	Returns the default home directory to assign to new users in the currently selected zone. The value can be a string that defines the path or include variables.
defaultshell	Returns the default shell to assign to new users in the currently selected zone. The value can be a string that defines the shell or include variables.
description	Returns the description of the zone. If the currently selected zone is a computer role, this field returns the Active Directory description attribute for the msds-AzScope object.
dn	Returns the zone's distinguished name. If the currently selected zone is a computer role, this field returns the Active Directory distinguished name attribute of the msds-AzScope object.
gidnext	Returns the next GID to use when auto-assigning GID numbers to new groups in the currently selected zone.
gidreserved	Returns the GID number or range of numbers (1-100) that are reserved in the currently selected zone.
groupname	Returns the default group name used for new groups in the currently selected zone. You can only return the value for this field if the current zone is a hierarchical zone.
modifyTime	Returns the time and date this zone was last modified (in generalized time format).
nisdomain	Returns the name of the NIS domain if it has been set. The default value is the zone name.
parent	Returns the distinguished name (DN) of the parent zone for the currently selected zone. You can only return the value for this field if the current zone is a hierarchical zone. You can use the option -raw with this field to return the parentLink attribute in the raw *Guid@Domain* format.
schema	Returns the schema used in this zone, for example, std.
sfudomain	Returns the Windows domain name for the SFU zone. Only use this argument if the current zone is a Service for UNIX (sfu) zone.
sid2iddomainmap	Returns a comma-separated key value pairs string. If an empty string is returned, that means that there's no domain ID mapping for the selected zone.
tenantid	Returns the tenant ID of the cloud instance associated with the selected zone.

type	Returns the type of the currently selected zone. For example, this field returns classic3 or classic4 for a classic zone or tree for a hierarchical zone.
uidnext	Returns the next UID to use when auto-assigning UID numbers to new users in the currently selected zone.
uidreserved	Returns the UID number or range of numbers (1-100) that are reserved in the currently selected zone.
username	Returns the default user name used for new users in the zone. You can only return the value for this field if the current zone is a hierarchical zone.

Examples

get_zone_field type

This example returns the zone type:

tree

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the zone:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.
- select_zone retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone computer:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

get_zone_group_field

Use the get_zone_group_field command to return the value for a specified field from the currently selected zone group stored in memory. The get_zone_group_field command does *not* query Active Directory for the zone group. If you change field values using ADEdit without saving the zone group to Active Directory, the field value you retrieve using get_zone_group_field won't match the same field value for the zone group stored in Active Directory.

Zone Type

Classic and hierarchical

Syntax

get_zone_group_field field

Abbreviation

gzgf

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the case-sensitive name of the field whose value to retrieve. The possible values are: **addn**: Returns the distinguished name of the Active Directory group object for the zone group. For example, if the group object is created in the default Users container, this field might return a path similar to CN=pubsteam,CN=Users,DC=ajax,DC=org. **createTime**: Returns the time and date this zone group was created (in generalized time format). **dn**: Returns the distinguished name of the service connection point for the zone group. If the zone is a Services for UNIX (sfu) zone, no value is returned for this field. **gid**: Returns the numeric identifier for the group. **modifyTime**: Returns the time and date this zone group was last modified (in generalized time format). **name**: Returns the group name. **required**: Returns 1 if the zone group is required for members in this zone, or 0 if the group is not required. Users assigned to a required group cannot remove the group from their active set of groups. You can also specify AIX extended attributes as the field to get an extended attribute value for a group. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be retrieved by specifying aix.admin as the field.

Return Value

This command returns a field value. The data type depends on the field specified.

Examples

The following example returns the group name.

```
get_zone_group_field name
```

```
padmins
```

If the current group is on AIX, you can get AIX group extended attributes and values. For example, to find out if the current group is an administrative group, you can get the admin extended attribute:

```
get_zone_group_field aix.admin
```

```
true
```

Related Commands

Before you use this command, you must have a currently selected zone group stored in memory. The following commands enable you to view and manage the zone groups:

- list_zone_groups lists to stdout the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

get_zone_groups

Use the get_zone_groups command to check Active Directory and return a Tcl list of zone groups defined within the currently selected zone. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed. Use list_zone_groups to output the list to stdout.

Zone Type

Classic and hierarchical

Syntax

```
get_zone_groups
```

Abbreviation

gzg

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of zone groups defined in the currently selected zone. Each entry in the list is the user principal name (UPN) of a group that you can use to look up that group.

Examples

```
get_zone_groups
```

This example returns the list of zone groups: poweradmins@acme.com auditors@acme.com

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- list_zone_groups lists to stdout the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the currently selected zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

get_zone_nss_vars

Use the get_zone_nss_vars command to return a Tcl list containing the NSS substitution variables for the currently selected zone stored in memory. This command only works on hierarchical zones and won't return a value for other zone types.

The get_zone_nss_vars command does *not* query Active Directory for this zone. If you change the variables using set_zone_field without saving the zone Active Directory, the variable you retrieve using get_zone_nss_vars won't match the same field variable for the zone stored in Active Directory.

Zone Type

Hierarchical only

Syntax

```
get_zone_nss_vars
```

Abbreviation

gznv

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of strings in the form "A=B".

Examples

```
get_zone_nss_vars
```

This example returns: NSSRANDCOUNT=32000 NSRANDFILE=/params/nssrand.seed

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone:

- `create_zone creates a new zone in Active Directory.
- `get_zones returns a Tcl list of all zones within a specified domain.
- `select_zone retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

get_zone_user_field

Use the get_zone_user_field command to return the value for a specified field from the currently selected zone user stored in memory. The get_zone_user_field command does *not* query Active Directory for the zone user. If you change field values using ADEdit without saving the zone user to Active Directory, the field value you retrieve using get_zone_user_field won't match the same field value for the zone user stored in Active Directory.

Zone Type

Classic and hierarchical

Syntax

```
get_zone_user_field field
```

Abbreviation

gzuf

Options

This command takes no options.

Arguments

This command takes the following required argument:

field (string type)

Specifies the case-sensitive name of the field whose value to retrieve.

Argument values

- **addn**: Returns the distinguished name of the Active Directory user object for the zone user. For example, if the user object is created in the default Users container, this field might return a path similar to CN=amy.adams,CN=Users,DC=ajax,DC=org.
- **createTime**: Returns the time and date this zone user was created.
- **dn**: Returns the distinguished name of the service connection point for the zone user. If the zone is a Services for UNIX (sfu) zone, no value is returned for this field.
- **enabled**: Returns 1 if the user is enabled, or 0 if the user is disabled. This field is only applicable for users in classic zones. All other zone types use roles.
- **foreign**: If the zone user comes from another forest, this field returns the user principal name of the zone user. Otherwise, this field returns no value.
- **gecos**: Returns information from the GECOS field.
- **gid**: Returns the primary group identifier (GID) for the user.
- **home**: the Returns user's home directory.
- **modifyTime**: Returns the time and date this zone user was last modified.
- **shell**: Returns the user's shell type.
- **uid**: Returns the numeric identifier for the user.
- **uname**: Returns the user name.

You can also specify AIX extended attributes as the field to get an extended attribute value for a zone user.

Return Value

This command returns a field value. The data type depends on the field specified.

Examples

The following example returns the current zone user's user name:

```
get_zone_user_field uname
adam
```

If the current zone user is on AIX, you can get extended attributes and values. For example:

```
select_zone_user aixu1@acme.com
get_zone_user_field aix.ttys
u1,u2,u3
```

Related Commands

Before you use this command, you must have a currently selected zone user stored in memory. The following commands enable you to view and select a zone user:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone.
- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

get_zone_users

Use the get_zone_users command to check Active Directory and return a Tcl list of zone users defined within the currently selected zone. If executed in a script, this command does not output its list tostdout, and no output appears in the shell where the script is executed. Use list_zone_users to output the list to stdout.

Zone Type

Classic and hierarchical

Syntax

```
get_zone_users [-upn]
```

Abbreviation

gzu

Options

This command takes the following option:

-upn	Optional. Returns user names in user principal name (UPN) format rather than the default sAMAccount@domain format.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of zone users defined in the currently selected zone. By default, users are listed by sAMAccountName@domain. You can use the -upn option to return users listed by user principal name (UPN). If a zone user is an orphan user—that is, its corresponding Active Directory user no longer exists—the user is listed by its security identifier (SID) instead of the sAMAccountName or user principal name.

Examples

get_zone_users

This example returns the list of users: adam.avery brenda.butler chris.carter

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone user:

- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

get_zones

Use the get_zones command to check Active Directory and return a Tcl list of zones within a specified domain. Note that this does not include computer-specific override zones or computer roles.

Zone Type

Classic and hierarchical

Syntax

get_zones domain

Abbreviation

gz

Options

This command takes no options.

Arguments

This command takes the following argument:

domain	string	Required. Specifies the name of the domain for which to return zones.

Return Value

This command returns a Tcl list with the distinguished name for each zone in the specified domain.

Examples

```
get_zones acme.com
```

This example returns the list of zones in the acme.com domain:

```
CN=childzone1,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com CN=childzone2,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com
CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com
```

Related Commands

The following commands perform actions related to this command:

- create_zone creates a new zone in Active Directory.
- select_zone retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

getent_passwd

Use the getent_passwd command to return a Tcl list of local UNIX users that are defined in the /etc/passwd file on the ADEdit host computer. If the local host is joined to an Active Directory domain, the command also returns information for the Active Directory users who have a profile in the joined domain and zone.

Zone Type

Not applicable

Syntax

```
getent_passwd
```

Abbreviation

```
gep
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a Tcl list of /etc/passwd file entries with all user profile attributes.

Examples

getent_passwd

This example returns the contents of the local /etc/passwd file:

```
{root x 0 0 root /root /bin/bash}{bin x 1 1 bin /bin /sbin/nologin}{daemon x 2 2 daemon /sbin /sbin/nologin}{adm x 3 4 adm /var/adm /sbin/nologin}{lp x 4 7 lp /var/spool/lpd /sbin/nologin}{sync x 5 0 sync /sbin /bin/sync}{shutdown x 6 0 shutdown /sbin /sbin/shutdown}
```

Related Commands

The following command performs actions related to this command:

- get_pwnam searches the /etc/passwd file for a UNIX user name and, if found, returns a Tcl list of the profile attributes associated with the user.

guid_to_id

Use the guid_to_id command to specify a globally unique identifier (GUID) for a user or group and returns a UID or GID that uses the Apple methodology for automatically generated unique identifiers.

Zone Type

Not applicable

Syntax

```
guid_to_id guid
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following argument:

guid	string	Required. Specifies the globally unique identifier for a user or group.

Return Value

This command returns UID or GID for the user or group generated using the Apple mechanism for automatically generating identifiers.

Examples

```
guid_to_id 763ddbc8-44cc-4a79-83aa-abc899b46aba
```

This example returns the UID for the user associated with the specified globally unique identifier:

```
1983765448
```

Related Commands

The following command performs actions related to this command:

- principal_to_id returns a unique UID or GID based on either the Apple methodology or the Delinea Auto Zone methodology for generating numeric identifiers.
- sid_to_uid converts a user's security identifier to a numeric identifier (UID).

help

Use the help command to return information about one or more ADEdit commands. It's followed by a command pattern that is either the name of a single ADEdit command or a string with wild cards that specifies multiple possible commands. The command pattern can also be a command abbreviation.

The command pattern wild cards are:

- ? for a single character
- * for multiple characters

Zone Type

Not applicable

Syntax

```
help command_pattern
```

Abbreviation

```
h
```

Options

This command takes no options.

Arguments

This command takes the following argument:

command_pattern	string	Required. Specifies the name of one or more ADEdit commands for which to return information. You can specify a command name, command shortcut or use the ? and * wild cards to specify a single character or multiple characters respectively.

Return Value

This command returns information for the specified command or commands. If there's no match for the *command_pattern* you specify, the command returns nothing.

Examples

```
help explain_sd
```

This example returns information for the explain_sd command.

```
help ?et*
```

This example returns information for the ADEdit commands that start with get or set, such as get_zones, get_zone_field, set_zone_field, and set_role_field.

Related Commands

None.

is_dz_enabled

Use this command to check whether authorization is enabled in a currently selected classic zone.

Zone Type

Classic only

Syntax

```
is_dz_enabled
```

Abbreviation

```
idze
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns 1 if authorization is enabled in a classic or 0 if authorization is not enabled.

Examples

```
create_zone classic4 cn=c125,cn=zones,dc=test,dc=net
select_zone cn=c125,cn=zones,dc=test,dc=net
is_dz_enable
0
manage_dz -on
is_dz_enable
1
```

This code example creates a new classic zone, checks that authorization is disabled by default, then enables authorization for the zone.

Related Commands

The following command performs actions related to this command:

- `manage_dz` enables and disables authorization in classic4 zones.

joined_get_user_membership

Use the `joined_get_user_membership` command to have `adclient` query Active Directory for a list of groups that a specified user belongs to in the domain to which ADEdit's host computer is joined. If the `adclient` query returns groups, this command returns those groups in a Tcl list.

Because this command queries Active Directory through `adclient`, the query might use the `adclient` cache instead of connecting directly to Active Directory. The `adclient` cache isn't guaranteed to be updated with ADedit activity. Therefore, you might need to execute the Delinea UNIX command `adflush` before using `joined_get_user_membership` to ensure you get the most up-to-date results.

Zone Type

Not applicable

Syntax

```
joined_get_user_membership user_UPN
```

Abbreviation

jgum

Options

This command takes no options.

Arguments

This command takes the following argument:

user_UPN	string	Required. Specifies the user principal name (UPN) of the user to check for group membership.

Return Value

This command returns a Tcl list of groups.

Examples

```
joined_get_user_membership liz.lemon@acme.com
```

This example returns group membership for liz.lemon in the joined domain:

```
acme.com/Users/Domain Users
```

Related Commands

The following commands performs actions related to this command:

- joined_user_in_group checks Active Directory through adclient to see if a user is in a group.
- get_effective_groups returns a Tcl list of groups a user belongs to.
- get_group_members returns a Tcl list of members in a group.

joined_name_to_principal

Use the joined_name_to_principal command have adclient query Active Directory for a UNIX name of a specified user. If the specified user is found, the command returns the associated Active Directory user name in the format of sAMAccountName@domain. The command can also optionally return the user principal name (UPN) of the user. This command works only for users within the domain to which ADEdit's host computer is joined through adclient.

Zone Type

Not applicable

Syntax

```
joined_name_to_principal [-upn] UNIX_name
```

Abbreviation

jntp

Options

This command takes the following option:

-upn	Returns the user's Active Directory name in user principal name (UPN) format.

Arguments

This command takes the following argument:

UNIX_name	string	Required. Specifies the UNIX name of a user to look for in Active Directory.

Return Value

This command returns the sAMAccountName@domain form of the user name if the user is found in Active Directory. If you specify the -upn option, this command returns the UPN form of user name.

Examples

```
joined_name_to_principal -upn adam
```

This example returns the sAMAccountName@domain for the UNIX user adam:

```
adam.avery@acme.com
```

Related Commands

The following commands performs actions related to this command:

- principal_to_dn searches Active Directory for a user principal name (UPN) and, if found, returns the corresponding DN.
- dn_to_principal searches Active Directory for a distinguished name and, if found, returns the corresponding UPN.
- principal_from_sid searches Active Directory for a security identifier (SID) and returns the security principal associated with the SID.

joined_user_in_group

Use the joined_user_in_group command to have adclient query Active Directory to see if a specified user belongs to a specified group. This command works only for users and groups within the domain to which ADEdit's host computer is joined through adclient.

Because this command queries Active Directory throughadclient, the query might use adclient's cache rather than connect directly to Active Directory. The adclient cache isn't guaranteed to be updated with ADedit activity. Therefore, you might need to execute the Delinea UNIX command adflush before using joined_user_in_group to ensure you get the most up-to-date results.

Zone Type

Not applicable

Syntax

```
joined_user_in_group user_UPN group_UPN
```

Abbreviation

```
jug
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

user_UPN	string	Required. Specifies the user principal name (UPN) of the user for which you want to check group membership.
group_UPN	string	Required. Specifies the UPN of the group for which you want to check user membership.

Return Value

This command returns 1 if the user is a member of the group, or 0 if the user is not a member of the group.

Examples

```
joined_user_in_group martin.moore@acme.com poweradmins@acme.com
```

This example returns 1 because martin.moore is a member of the poweradmins group.

Related Commands

The following commands performs actions related to this command:

- joined_get_user_membership uses adclient to return a Tcl list of groups that a user belongs to.
- get_effective_groups checks Active Directory directly and returns a Tcl list of groups a user belongs to.
- get_group_members checks Active Directory and returns a Tcl list of members in a group.

list_dz_commands

Use the list_dz_commands command to check Active Directory and return a list of UNIX command objects defined within the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_dz_commands to return a Tcl list.

You can only use the list_dz_commands command to return UNIX command data for classic4 and hierarchical zones.

Zone Type

Classic and hierarchical

Syntax

```
list_dz_commands
```

Abbreviation

```
lsdzc
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of UNIX commands defined in the currently selected zone. Each entry in the list contains the following fields, separated

by colons (:):

- The name of the UNIX command followed by a slash (/) and the name of the zone where the command is defined.
- The properties of the command.
- Text describing the command.

Examples

```
list_dz_commands
```

This example returns commands in the following format:

```
root_any/global : * form(0) dzdo_runas(root) flags(16) : Run any command as root
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command:

- `get_dz_commands` returns a Tcl list of UNIX commands in the current zone.
- `new_dz_command` creates a new UNIX command and stores it in memory.
- `select_dz_command` retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- `delete_dz_command` deletes the selected command from Active Directory and from memory.
- `get_dzc_field` reads a field value from the currently selected command.
- `save_dz_command` saves the selected command with its current settings to Active Directory.
- `set_dzc_field` sets a field value in the currently selected command.

list_local_groups_profile

Use the `list_local_groups_profile` command to display a list of local UNIX and Linux group profiles that are defined in the current zone.

Zone Type

Hierarchical only.

Syntax

```
list_local_groups_profile
```

Abbreviation

```
lslgp
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to `stdout` of the local UNIX and Linux group profiles that are defined in the current zone. Each profile contains the group name, GID, members, and profile flag value.

Examples

The following example returns a local group profile list.

```
list_local_groups_profile lam_grp1:3001:lam_usr1:1 lam_grp2:3002:lam_usr2:1 lam_grp3:3003:lam_usr3:3
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

list_local_users_profile

Use the list_local_users_profile command to display a list of local UNIX and Linux user profiles that are defined in the current zone.

Zone Type

Hierarchical only.

Syntax

```
list_local_users_profile
```

Abbreviation

```
lslup
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of the local UNIX and Linux user profiles that are defined in the current zone. Each profile contains the user name, UID, primary GID, GECOS, home directory, shell, and profile flag value.

Examples

The following example returns a local user profile list.

```
list_local_users_profile
```

```
lam_usr1:2001:2001:lam usr1:/home/lam_usr1:/bin/bash:1
```

lam_usr2:2002:2002:lam usr2:/home/lam_usr2:/bin/bash:2

lam_usr3:2003:2003:lam usr3:/home/lam_usr3:/bin/bash:3

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

list_nis_map

Use the list_nis_map command to return a list of all map entries within the currently selected NIS map. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_nis_map to return a Tcl list of NIS map entries.

Zone Type

Not applicable

Syntax

list_nis_map

Abbreviation

lsnm

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of the map entries for the currently selected NIS map. Each map entry in the list contains the following fields separated by colons (:):

- The key
- The instance number of the key
- The value

Examples

list_nis_map

This example returns map entries similar to the following:

Finance:1:Hank@acme.com,jane@acme.com,joe@acme.com

Mktg:1:Mike@acme.com,Sue@acme.com

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select a NIS map:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps returns a list to stdout of all NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

- add_map_entry or add_map_entry_with_comment adds a map entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map_with_comment lists to stdout the map entries in the currently selected NIS map.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

list_nis_map_with_comment

Use the list_nis_map_with_comment command to return a list of all map entries for the currently selected NIS map and includes the entries' comment. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed.

The command does not return a Tcl list back to the executing script. Use get_nis_map or get_nis_map_with_comment to return a Tcl list of NIS map entries for parsing or further processing within the script.

Zone Type

Not applicable

Syntax

list_nis_map_with_comment

Abbreviation

lsnmwc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of the map entries for the currently selected NIS map. Each map entry in the list contains the following fields separated by colons ':' :

- The key
- The instance number of the key
- The value
- The comment

Examples

list_nis_map_with_comment

This example returns map entries similar to the following:

Finance:1:Hank@acme.com,jane@acme.com,joe@acme.com:Finance dept staff

Mktg:1:Mike@acme.com,Sue@acme.com:Marketing dept staff

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select a NIS map:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout the NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

- add_map_entry or add_map_entry_with_comment adds a map entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map lists to stdout the map entries in the currently selected NIS map.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

list_nis_maps

Use the list_nis_maps command to check Active Directory and return a list of NIS maps defined in the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_nis_maps to return a Tcl list.

Zone Type

Not applicable

Syntax

list_nis_maps

Abbreviation

lsnms

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of NIS maps defined in the currently selected zone.

Examples

list_nis_maps

This example returns the list of NS maps for the zone:

Aliases

Printers

Services

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select a NIS map:

- get_nis_maps returns a Tcl list of NIS maps in the currently selected zone.
- list_nis_maps lists to stdout the NIS maps in the currently selected zone.
- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

- add_map_entry or add_map_entry_with_comment adds a map entry to the currently selected NIS map.
- delete_map_entry removes an entry from the currently selected NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout the map entries in the currently selected NIS map.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

list_pam_apps

Use the list_pam_apps command to check Active Directory and return a list of PAM application rights defined in the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_pam_apps to return a Tcl list.

You can only use the list_pam_apps command to return PAM application rights for classic4 and hierarchical zones.

Zone Type

Classic and hierarchical

Syntax

list_pam_apps

Abbreviation

lspa

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of PAM application rights defined in the currently selected zone. Each entry contains the following fields, separated by colons : :

- The name of the PAM access right followed by a slash (/) and the zone in which the PAM access right is defined.
- The name of one or more PAM applications to which the right applies.
- Text describing the PAM application object.

Examples

list_pam_apps

This example returns a list of PAM application access rights for the selected zone (the following is a subset of the default predefined rights):

```
dzssh-all/global : dzssh-* : All of ssh services
dzssh-exec/global : dzssh-exec : Command execution
dzssh-scp/global : dzssh-scp : scp
dzssh-sftp/global : dzssh-sftp : sftp
dzssh-shell/global : dzssh-shell : Terminal tty/pty
dzssh-tunnel/global : dzssh-tunnel : Tunnel device forwarding
dzssh-X11-forwarding/global : dzssh-x11-forwarding : X11 forwarding
login-all/global : * : Predefined global PAM permission. Do not delete.
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a PAM application object:

- get_pam_apps returns a Tcl list of PAM applications in the current zone.
- new_pam_app creates a new PAM application and stores it in memory.
- select_pam_app retrieves a PAM application from Active Directory and stores it in memory.

After you have a PAM application object stored in memory, you can use the following commands to work with that PAM application:

- delete_pam_app deletes the selected PAM application from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application.
- save_pam_app saves the selected PAM application with its current settings to Active Directory.
- set_pam_field sets a field value in the currently selected PAM application.

list_pending_zone_groups

Use the list_pending_zone_groups command to check Active Directory and return a list of pending import groups for the currently selected zone. Pending import groups are group profiles that have been imported from Linux or UNIX computers, but not yet mapped to any Active Directory group. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_pending_zone_groups to return a Tcl list.

Zone Type

Classic and hierarchical

Syntax

list_pending_zone_groups

Abbreviation

lpzg

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of pending import groups for the currently selected zone. Each entry in the list contains the following fields, separated by colons (:) :

- Distinguished name (DN) of the pending import group as it is stored in Active Directory. The distinguished name for each pending import group includes a prefix that consists of "PendingGroup" and the globally unique identifier (GUID) for the group.
- UNIX group name.
- Numeric group identifier (GID).

Examples

```
list_pending_zone_groups
```

This example returns the list of groups similar to this:

```
CN=PendingGroup_573135e7-edd9-46b9-9cbd-c839570a90c8,CN=Groups, CN=bean_pz,CN=Zones,CN=Acme,DC=win2k3,DC=test:root:0   CN=PendingGroup_7878065a-4d2f-4749-8f3b-6ffe24303f6a,CN=Groups, CN=bean_pz,CN=Zones,CN=Acme,DC=win2k3,DC=test:unixgrp:5000
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following command performs actions related to this command:

- get_pending_zone_groups returns a Tcl list of the pending import groups in the current zone.

list_pending_zone_users

Use the list_pending_zone_users command to check Active Directory and return a list of pending import users for the currently selected zone. Pending import users are user profiles that have been imported from Linux or UNIX computers, but not yet mapped to any Active Directory user. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_pending_zone_users to return a Tcl list.

Zone Type

Classic and hierarchical

Syntax

```
list_pending_zone_users
```

Abbreviation

```
lpzu
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of pending import users for the currently selected zone. Each entry in the list contains the following fields, separated by colons (:) :

- Distinguished name (DN) of the pending import user as it is stored in Active Directory. The distinguished name for each pending import user includes a prefix that consists of "PendingUser" and the globally unique identifier (GUID) for the user.
- UNIX user name.
- Numeric user identifier (UID).
- Numeric primary group identifier (GID).

- Personal information from the GECOS field.
- Home directory.
- Default login shell.

Examples

list_pending_zone_users

This example returns the list of groups similar to this:

CN=PendingUser_09024f3a-6abc-4666-a127-722f9fe0e0bf,CN=Users,CN=finance, CN=Zones,CN=Acme,DC=win2k3,DC=test:root:0:0:root:/root:/bin/bash: CN=PendingUser_0b9fe038-1325-438f-8529-cb190ab5914a,CN=Users,CN=finance, CN=Zones,CN=Acme,DC=win2k3,DC=test:bean:6001:5000:bean.zhang:/home/bean:/bin/bash:

Related Commands

The following command performs actions related to this command:

- get_pending_zone_users returns a Tcl list of the pending import users in the current zone.

list_role_assignments

Use the list_role_assignments command to check Active Directory and return a list of role assignments defined within the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_role_assignments to return a Tcl list.

If you do not specify an option, the command returns the current users and groups in the zone with a role assignment using the default sAMAccount@domain format.

You can only use the list_role_assignments command to return role assignments for classic4 and hierarchical zones.

Zone Type

Classic and hierarchical

Syntax

list_role_assignments [-upn] [-visible] [-user] [-group] [-invalid]

Abbreviation

lsra

Options

This command takes the following options:

-upn	Optional. Returns user names in user principal name (UPN) format rather than the default sAMAccount@domain format.
-visible	Returns a list to stdout of the visible role assignments in the zone. Use this option if you only want to return role assignments for the roles that are identified as visible. This option is only applicable in hierarchical zones.
-user	Returns a list to stdout of the current users in the zone with a role assignment. Use this option if you only want to return valid users with a role assignment.
-group	Returns a list to stdout of the current groups in the zone with a role assignment. Use this option if you only want to return valid groups with a role assignment.
-invalid	Returns a list to stdout of any invalid role assignments in the zone. A role assignment is invalid if it specifies a group or user that no longer exists. Use this option if you only want to return invalid role assignments.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of role assignments defined in the currently selected zone. Each entry in the list provides the following information:

- The user or group to whom the role assignment applies by sAMAccount@domain name or user principal name.
- The name of the role assigned followed by a slash (/) and the zone where the role is defined.

Examples

```
>bind pistolas.org

>select_zone “cn=northamerica,cn=zones,ou=acme,dc=pistolas,dc=org”

>list_role_assignments
```

This example returns the role assignments for the northamerica zone: Domain Users@pistolas.org: Window Login/northamerica adm-sf@pistolas.org: UNIX Login/northamerica rey@pistolas.org: UNIX Login/northamerica maya@pistolas.org: SQLAdmin/northamerica

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a role assignment:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the currently selected role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- set_role_assignment_field sets a field value in the currently selected role assignment.
- write_role_assignment saves the selected role assignment to a file.

list_role_rights

Use the list_role_rights command to return a list of all UNIX commands and PAM application rights set within the currently selected role. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script.

The list_role_rights command does *not* query Active Directory for the role. If you change commands or PAM applications using ADEdit without saving the role to Active Directory, commands and PAM applications you retrieve using list_role_rights won't match those stored in Active Directory.

You can only use list_role_rights to return role rights for classic4 and hierarchical zones.

Zone Type

Classic and hierarchical

Syntax

```
list_role_rights
```

Abbreviation

```
lsrr
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of the PAM application and UNIX command rights that are defined for the currently selected role.

Each entry lists the name of the application or command right, the attributes of the application or command, and any descriptive text.

Examples

```
list_role_rights
```

This example returns the list of PAM application and UNIX command rights:

```
dzssh-all/northamerica : dzssh-exec : Command execution   login-all/seattle : * : Predefined global PAM permission. Do not delete.   cron-exec/seattle : cron form(0) dzdo_runas(admin) flags(16)
;
```

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select a role:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles returns a list of all roles in the currently selected zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_role adds a UNIX command right to the current role.
- add_pamapp_to_role adds a PAM application right to the current role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM application rights associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the current role.
- remove_command_from_role removes a UNIX command from the current role.
- remove_pamapp_from_role removes a PAM application from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the current role.

list_roles

Use the list_roles command to check Active Directory and return a list of roles defined in the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_roles to return a Tcl list.

You can only use list_roles to return role information for classic4 and hierarchical zones.

Zone Type

Classic and hierarchical

Syntax

```
list_roles
```

Abbreviation

```
lsr
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of roles defined in the currently selected zone.

Examples

```
list_roles
```

This example returns the list of roles for the zone:

```
Rescue - always permit login
listed
scp
sftp
UNIX Login
Windows Login
winscp
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a role:

- get_roles returns a Tcl list of roles in the current zone.
- new_role creates a new role and stores it in memory as the currently selected role.
- select_role retrieves a role from Active Directory and stores it in memory as the selected role.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_role adds a UNIX command right to the current role.
- add_pamapp_to_role adds a PAM application right to the current role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM application rights associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the current role.
- list_role_rights returns a list of all UNIX command and PAM application rights associated with the current role.
- remove_command_from_role removes a UNIX command from the current role.
- remove_pamapp_from_role removes a PAM application from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the current role.

list_rs_commands

Use the list_rs_commands command to print a list of the restricted shell commands that are defined for the currently selected zone. This command retrieves information from Active Directory and to returns the list of restricted shell commands to stdout. If you want to return a Tcl list of restricted shell commands, use get_rs_commands.

Zone Type

Classic only

Syntax

```
list_rs_commands
```

Abbreviation

lsrsc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list of restricted shell commands for the currently selected zone.

Examples

list_rs_commands

This command returns the list of restricted shell commands and attributes similar to this:

rseid1/c123 : id form(0) dzsh_runas($) umask(77) path(USERPATH) flags(0) : rseid2/c123 : id2 form(0) dzsh_runas($) pri(1) umask(77) path(USERPATH) flags(0) : id2

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the restricted shell commands:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- new_rs_command creates a new restricted shell command and stores it in memory.
- save_rs_command saves the selected restricted shell command with its current settings to Active Directory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with its fields:

- get_rsc_field reads a field value from the current restricted shell command.
- set_rsc_field sets a field value in the current restricted shell command.

list_rs_envs

Use the list_rs_envs command to check Active Directory and print a list of restricted shell environments defined within the currently selected zone to stdout. Use the get_rs_envs command to return a Tcl list.

Zone Type

Classic only

Syntax

list_rs_envs

Abbreviation

lsrse

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command prints the list of restricted shell environments to stdout. It has no return value.

Examples

list_rs_envs

This example displays the list of restricted shell environments.

restrict_env1

restrict_env2

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

list_zone_computers

Use the list_zone_computers command to check Active Directory and return a list of zone computers defined within the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_zone_computers to return a Tcl list.

Zone Type

Classic and hierarchical

Syntax

list_zone_computers

Abbreviation

lszc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of zone computers defined in the currently selected zone. Each zone computer entry includes the following fields, separated by colons (:) :

- User principal name (UPN) of the computer.
- Number of CPUs in the computer and the version of Delinea software installed on the computer.
- Name of the computer in DNS.

Examples

```
list_zone_computers
```

This example returns the list of computers similar to this:

```
printserv$@acme.com:cpus (1) agentVersion (CentrifyDC 5.0.0): printserv.acme.com
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone computer:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

list_zone_groups

Use the list_zone_groups command to check Active Directory and return a list of zone groups defined in the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_zone_groups to return a Tcl list.

Zone Type

Classic and hierarchical

Syntax

```
list_zone_groups
```

Abbreviation

```
lszg
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of zone groups defined in the currently selected zone. Each entry in the list contains the following fields, separated by colons (:) :

- User principal name of the zone group as it is stored in Active Directory.
- UNIX group name.
- Numeric group identifier (GID).
- The string "Required" if the "Users are required to be members of this group" option is set for the group.

Examples

```
list_zone_groups
```

This example returns the list of groups similar to this:

```
sf-admins@pistolas-org:sfadmins:10F24
sf-apps@pistolas.org:sf-apps:2201
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- get_zone_groups returns a Tcl list of the Active Directory names of the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the currently selected zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

list_zone_users

Use the list_zone_users command to check Active Directory and return a list of zone users defined in the currently selected zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use get_zone_users to return a Tcl list.

Zone Type

Classic and hierarchical

Syntax

```
list_zone_users [-upn]
```

Abbreviation

```
lszu
```

Options

This command takes the following option:

-upn	Optional. Returns user names in user principal name (UPN) format rather than the default sAMAccount@domain format.

Arguments

This command takes no arguments.

Return Value

This command returns a list to stdout of zone users for the currently selected zone. Each entry in the list contains the following user profile fields separated by colons (:) :

- sAMAccountName@domain or the UPN of the zone user as it is stored in Active Directory.

  If the Active Directory user no longer exists for a zone user, the command returns the security identifier (SID) of the orphan user.
- UNIX user name.
- Numeric user identifier (UID).
- Numeric identifier for the user's primary group (GID).

  If the GID has the number 2147483648 (which is 80000000 hex) it means that the UID is being used as the GID. (This can occur in hierarchical zones.)
- Personal information from the GECOS field.
- The user's home directory.
- The user's default login shell.
- Whether the user is enabled or disabled (in classic zones only).

Examples

```
list_zone_users
```

This example returns the list of users similar to this:

```
adam.avery@acme.com:adam:10001:10001:%{u:samaccountname}:%{home}/%{user}:%{shell}:
ben.brown@acme.com:brenda:10002:10002:%{u:samaccountname}:%{home}/%{user}:%{shell}:
chris.cain@acme.com:chris:10003:10003:%{u:samaccountname}:%{home}/%{user}:%{shell}:
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone users:

- get_zone_users returns a Tcl list of the Active Directory names of zone users in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

manage_dz

Use the manage_dz command to enable or disable authorization in classic zones. In classic zones, authorization-related features are disabled by default, and the authorization store that is required for managing rights, roles, and restricted environment is not available in Active Directory.

To enable authorization in classic zones using ADEdit, you can run the manage_dz``-on command. This command creates the authorization store if it does not exist, and sets the zone property that enables privilege elevation service features.

To disable authorization in a classic zone, you can run the manage_dz –off command. Running this command disables authorization services. The command does not remove any existing authorization data from Active Directory.

Zone Type

Classic only

Syntax

manage_dz [-on l-off]

Abbreviation

mnz

Options

This command takes the following options:

-on	Enables authorization for the currently selected zone and creates the authorization data store if it not currently defined in Active Directory.
-off	Disables authorization for the currently selected zone. This option does not remove any data from the authorization data store if it currently exists.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
create_zone classic4 cn=c125,cn=zones,dc=ross,dc=net
select_zone cn=c125,cn=zones,dc=ross,dc=net
is_dz_enable
0
manage_dz -on
is_dz_enable
1
```

This code example creates a zone, checks that authorization is disabled by default, then enables authorization for the zone.

Related Commands

The following command performs actions related to this command:

is_dz_enabled checks whether authorization is currently enabled for a zone.

move_object

Use the move_object command to move the selected object to the specified location. The new location must be in the same domain. You cannot use this command to move an object to another domain. You do not need to save the object after moving it.

Zone Type

Not applicable

Syntax

move_object destination

Abbreviation

mvo

Options

This command takes no options.

Arguments

This command takes the following argument:

destination	string	Required. Specifies the distinguished name of the new location.

Return Value

This command returns nothing if it runs successfully.

Example

The following commands move the ApacheAdmins group from the Groups container in the Global zone to the Groups container in the US zone.

```
select_object “cn=ApacheAdmins@demo.test,cn=Groups,cn=Global,cn=Zones,CN=Acme,dc=demo,dc=test”  mvo “cn=Groups,cn=US,cn=Zones,ou=Acme,dc=demo,dc=test”
```

Related Commands

The following command performs actions related to this command:

- select_object selects the object you want to move.

new_dz_command

Use the new_dz_command command to create a new UNIX command object for the current zone and sets the new command as the currently selected command in memory. The new command has no field values set. The new_dz_command does *not* save the new command to Active Directory. To save the UNIX command, you must first set at least the “command” field using set_dzc_field, then use save_dz_command. If you don’t save a new UNIX command, it will disappear when you select a new command or when the ADEdit session ends.

You can only use the new_dz_command command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
new_dz_command _name_
```

Abbreviation

newdzc

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the name to assign to the new UNIX command.

Return Value

This command returns nothing if it runs successfully.

Examples

new_dz_command account_manager

This example creates a new UNIX command named account_manager in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select UNIX commands:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands returns a list of all UNIX commands in the currently selected zone.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- get_dzc_field reads a field value from the currently selected command.
- save_dz_command saves the selected command with its current settings to Active Directory.
- set_dzc_field sets a field value in the currently selected command.

new_local_group_profile

Use the new_local_group_profile command to create an object for a local UNIX or Linux group in the currently selected zone. After you create the group object, it is automatically selected for editing with the set_local_group_profile_field command. That is, you do not need to execute the select_local_group_profile command to select the new group prior to defining profile fields. After you create the new group, save it by executing the save_local_group_profile command.

When the group profile is complete and the profileflag field is set to 1 (enabled), the profile is added to /etc/group on each UNIX and Linux computer in the zone at the next local account refresh interval. A group profile must have the following fields (attributes) to be considered complete:

- A unique numeric identifier (GID).
- A group name.

See the *Administrator's Guide for Linux and UNIX* for more details about creating local group profiles.

Zone Type

Hierarchical only.

Syntax

new_local_group_profile group_name

Abbreviation

newlgp

Options

This command takes no options.

Arguments

This command takes the following argument:

group_name	string	Required. Specifies the UNIX name of the new local group to create in the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example shows a typical sequence of commands to create an object for the local UNIX or Linux group marketing in the currently selected zone. This command sequence creates a complete group profile, and sets the profile flag to 1 (enabled) so that the profile is added to /etc/group at the next local account update interval.

```
new_local_group_profile marketing
set_local_group_profile_field gid 3004
set_local_group_profile_field member lam_usr4
set_local_group_profile_field profileflag 1
save_local_group_profile
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

new_local_user_profile

Use the new_local_user_profile command to create an object for a local UNIX or Linux user in the currently selected zone. After you create the user object, it is automatically selected for editing with the set_local_user_profile_field command. That is, you do not need to execute the select_local_user_profile command to select the new user prior to defining profile fields. After you create the new user, save it by executing the save_local_user_profile command.

> **Note**:Unlike local groups, which are visible by default, you must explicitly assign local users to a visible role. If you do not assign a local user to a visible role, the user profile defined in the zone object is not updated in /etc/passwd on local computers. A predefined visible role for local users, local listed, is provided to make local users visible. After you create a local user profile, you must assign the local user to the local listed role, or to another visible role. You can use the select_role_assignment and new_role_assignment ADEdit commands to make role assignments.

When the user profile is complete and the profileflag field is set to 1 (enabled) or 2 (disabled), the profile is added to /etc/passwd on each UNIX and Linux computer in the zone at the next local account refresh interval.

A user profile must have the following fields (attributes) to be considered complete:

- A user name (the UNIX login name).
- A unique numeric user identifier (UID).
- The user's primary group profile numeric identifier (GID).
- The default home directory for the user.
- The default login shell for the user.

Note that the GECOS field is not required.

See the *Administrator's Guide for Linux and UNIX* for more details about creating local user profiles.

Zone Type

Hierarchical only.

Syntax

```
new_local_user_profile user_name
```

Abbreviation

```
newlup
```

Options

This command takes no options.

Arguments

This command takes the following argument:

user_name	string	Required. Specifies the UNIX name of the new local user to create in the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example shows a typical sequence of commands to create an object for the local UNIX or Linux user `lam_usr4` in the currently selected zone. This command sequence creates a complete user profile, sets the profile flag to `1` (enabled), and adds the user to the `local listed` role so that the profile is added to `/etc/passwd` at the next local account update interval.

```
new_local_user_profile lam_usr4
set_local_user_profile_field uid 2004
set_local_user_profile_field gid 2004
set_local_user_profile_field shell /bin/bash
set_local_user_profile_field home /home/lam_usr4
set_local_user_profile_field profileflag 1
save_local_user_profile
select_role_assignment local listed
new_role_assignment lam_usr4
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- `delete_local_group_profile` deletes a local UNIX or Linux group that has a profile defined in the current zone.
- `delete_local_user_profile` deletes a local UNIX or Linux user that has a profile defined in the current zone.
- `get_local_group_profile_field` displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current

zone.

- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

new_nis_map

Use the new_nis_map command to create a new NIS map for the current zone and set the new NIS map as the currently selected NIS map in memory. The new NIS map has no map entries.

The new_nis_map does *not* save the new NIS map to Active Directory. To save the new map, you must use save_nis_map. If you don't save a new NIS map, it will disappear when you select a new NIS map or when the ADEdit session ends.

Zone Type

Not applicable

Syntax

```
new_nis_map [-automount] map
```

Abbreviation

newnm

Options

This command takes the following option:

-automount	Specifies that the new NIS map is an automount map. For most NIS maps, the map name defines the type of map you are creating. For example, if you create a new NIS map with the name netgroup, it must be a NIS netgroup map and contain valid netgroup entries. However, you can specify any name for NIS automount maps. Use this option to identify automount maps that have a name other than automount.

Arguments

This command takes the following argument:

map	string	Required. Specifies the name of the new NIS map. For most NIS maps, the map name defines the type of map you are creating. For example, if you create a new NIS map with the name netgroup, it must be a NIS netgroup map and contain valid netgroup entries. For information about the type of NIS maps you can create, see the_Network Information Service Administrator's Guide_.

Return Value

This command returns nothing if it runs successfully.

Examples

The following command creates the NIS map “Printers” in the current zone.

```
new_nis_map Printers
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select NIS maps:

- get_nis_maps returns a Tcl list of NIS maps in the current zone.
- list_nis_maps lists to stdout the NIS maps in the current zone.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

*add_map_entryOr add_map_entry_with_comment adds an entry to the current NIS map stored in memory.

- delete_map_entry removes an entry from the current NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the current NIS map.
- get_nis_map_field reads a field value from the current NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout ` the map entries in the current NIS map.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

new_object

Use the new_object command to create a new Active Directory object and set the new object as the currently selected Active Directory object in memory. The new object has no field values set. The new_object command does *not* save the new object to Active Directory. To save the new object, you must use save_object. If you don't save a new object, it will disappear when you select a new object or when the ADEdit session ends.

The new_object command does not check to see if the new object conforms to Active Directory's expectations for the new object in the location you specify. Active Directory will report any errors when you try to save the object.

Zone Type

Not applicable

Syntax

```
new_object dn
```

Abbreviation

```
newo
```

Options

This command takes no options.

Arguments

This command takes the following argument:

dn	DN	Required. Specifies the distinguished name for the new object.

Return Value

This command returns nothing if it runs successfully.

Examples

```
new_object "ou=Acme,cn=Program Data,dc=acme,dc=com"
```

This example creates a new organizational unit Delinea in the container Program Data in the domain acme.com and stores it in memory as the currently selected Active Directory object.

Related Commands

The following commands enable you to view and select Active Directory objects:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects matching the specified search criteria.
- select_object retrieves an object with its attributes from Active Directory and stores it in memory.

After you have an object stored in memory, you can use the following commands to work with that object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_fieldsets a field value in the currently selected Active Directory object.

new_pam_app

Use the new_pam_app command to create a new PAM application right for the current zone and set the new PAM application as the currently selected PAM application in memory. The new PAM application has no field values set.

The new_pam_app does *not* save the new PAM application to Active Directory. To save the PAM application right, you must first set at least the "application" field using set_pam_field, then use save_pam_app. If you don't save a new PAM application, it will disappear when you select a new PAM application or when the ADEdit session ends.

You can only use the new_pam_app to create PAM application rights if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
new_pam_app name
```

Abbreviation

```
newpam
```

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the name to assign to the new PAM application access right.

Return Value

This command returns nothing if it runs successfully.

Examples

```
new_pam_app basic
```

This example creates a new PAM application access right named basic in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select PAM application rights:

- get_pam_apps returns a Tcl list of PAM application rights in the current zone.
- list_pam_apps lists to stdout the PAM application rights in the currently selected zone.
- select_pam_app retrieves a PAM application right from Active Directory and stores it in memory.

After you have a PAM application right stored in memory, you can use the following commands to work with that PAM application right:

- delete_pam_app deletes the selected PAM application right from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application right.
- save_pam_app saves the selected PAM application right with its current settings to Active Directory.
- set_pam_field sets a field value in the currently selected PAM application right.

new_role

Use the new_role command to create a new role for the current zone and set the new role as the currently selected role in memory. The new role has no field values set. The new_role command does *not* save the new role to Active Directory. To save the new role, you must use save_role. If you don't save a new role, it will disappear when you select another role or when the ADEdit session ends.

You can only use the new_role to create a role if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
new_role _name
```

Abbreviation

```
newr
```

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the name to assign to the new role.

Return Value

This command returns nothing if it runs successfully.

Examples

```
new_role customerservice
```

This example creates a new role named customerservice in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select roles:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_roleadds a UNIX command to the current role.
- add_pamapp_to_role adds a PAM application to the current role.
- delete_role deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications associated with the currently selected role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM application rights associated with the current role.
- remove_command_from_role removes a UNIX command from the current role.
- remove_pamapp_from_role removes a PAM application from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the currently selected role.

new_role_assignment

Use the new_role_assignment command to create a new role assignment for the current zone and set the new role assignment as the currently selected role assignment in memory. The new role assignment has no field values set.

The new_role_assignment command does *not* save the new role assignment to Active Directory. To save the role assignment, you must first set at least the "role" field using set_role_assignment_field, then use save_role_assignment. If you don't save a new role assignment, it will disappear when you select another role assignment or when the ADEdit session ends.

You can only use the new_role_assignment to create a role assignment if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
new_role_assignment userIAll AD usersIAll Unix users
```

Abbreviation

```
newra
```

Options

This command takes no options.

Arguments

This command takes the following argument:

user | All AD users | All Unix users	string	Required. Specifies the user or group to assign the role to. This argument can be a user principal name (UPN) or a sAMAccountName if you are assigning a role to an Active Directory user or group, a UNIX user name or UID if assigning the role to a local UNIX user, or the UNIX group name if you assigning the role to a local UNIX group. To assign a role to a local UNIX account, use the following format: oracle@localhost To assign the role to a domain user, use the following format: oracle@*domain.name* You can also specify "All AD users" to assign a selected role to all Active Directory users or "All Unix users" to assign the selected role to all local UNIX users. This argument is not supported if the selected zone is a classic4 zone.

Return Value

This command returns nothing if it runs successfully.

Examples

```
new_role_assignment adam.avery@acme.com
```

This example creates a new role assignment for adam.avery@acme.com in the current zone. You must set at least one role assignment field and an available time for the role to be effective.

The following example creates a new role assignment for the local UNIX user oracle in the current zone.

```
new_role_assignment oracle@localhost
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select role assignment to work with:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- list_role_assignments lists to stdout the role assignments in the current zone.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment's attributes, delete the role assignment, or save information for the role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the currently selected role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- set_role_assignment_field sets a field value in the currently selected role assignment.
- write_role_assignment saves the selected role assignment to a file.

new_rs_command

Use the new_rs_command command to create a new restricted shell command under the currently selected restricted shell environment and set the new restricted shell command as the currently selected restricted shell command in memory. The umask field for the new restricted shell command is set to a default value of 077 and default priority field (pri) is set to 0. For more information about restricted shell command fields, see the command description for get_rsc_field.

The new_rs_command command does not save the new restricted shell command to Active Directory. To store the new restricted shell command in Active Directory, you must use save_rs_command. If you don't save a new restricted shell command, it will disappear when you select another restricted shell command or when the ADEdit session ends.

You can only use the new_rs_command command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

new_rs_command name

Abbreviation

newrsc

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the name to assign to the new restricted shell command.

Return Value

This command returns nothing if it runs successfully.

Examples

new_rs_command rsc1

This example creates a new restricted shell command named rsc1 in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

new_rs_env

Use the new_rs_env command to create a new restricted shell environment for the current zone and set the new restricted shell environment as the currently selected restricted shell environment stored in memory. The new restricted shell environment has no field values set.

The new_rs_env command does not save the new restricted shell environment to Active Directory. To save the new restricted shell environment to Active Directory, you must use the save_rs_env command. If you don't save a new restricted shell environment, it will disappear when you select another restricted shell environment or when the ADEdit session ends.

You can only use the new_rs_env command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

new_rs_env name

Abbreviation

newrse

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the name to assign to the new restricted shell environment.

Return Value

This command creates a new restricted shell environment in the currently selected zone.

Examples

new_rs_envs rse3

This example creates a new restricted environment named rse3 in the current zone.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

new_zone_computer

Use the new_zone_computer command to create a new zone computer in the current zone and set the new zone computer as the currently selected zone computer in memory. The new zone computer has no field values set.

The new_zone_computer command does *not* save the new zone computer to Active Directory. To save the new zone computer, you must use save_zone_computer. If you don't save a new zone computer, it will disappear when you select another zone computer or when the ADEdit session ends.

The new_zone_computer command requires you to specify an Active Directory computer account name. If the computer name you specify is not found in Active Directory, the command does not create the zone computer.

Zone Type

Classic and hierarchical

Syntax

new_zone_computer sAMAccountName@domain

Abbreviation

newzc

Options

This command takes no options.

Arguments

This command takes the following argument:

sAMAccountName @domain	string	Required. Specifies the sAMAccountName of an Active Directory computer followed by @ and the domain name where the computer is located.

Return Value

This command returns nothing if it runs successfully.

Examples

new_zone_computer sales2$@acme.com

This example creates a new zone computer sales2@acme.com in the current zone. Note that Tcl syntax requires $@ to represent an actual ampersand symbol @. You could also enclose the argument in braces: {sales2@acme.com}.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the zone computers:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

new_zone_group

Use the new_zone_group command to create a new group in the current zone that is based on an existing Active Directory group. If the command is successful, the new zone group becomes the currently selected zone group stored in memory.

The new_zone_group command does not set any field values or save the new zone group to Active Directory. Before you can save the new zone group, you must first set at least one field for the new zone group using the set_zone_group_field command. You can then save the zone group profile using the save_zone_group command.

> **Note**:If the currently selected zone is a classic zone, you must set all fields for the new zone group before saving the group profile.

If you don't save a new zone group, it will disappear when you select another zone group or end the ADEdit session.

The new_zone_group command requires you to specify an Active Directory group name. The command will search for the group first by the supplied UPN in the specified domain, then by the **sAMAccountname** in the specified domain, then by the supplied UPN in any bound domain. If the group name cannot be found, the new zone group is not created.

Zone Type

Classic and hierarchical

Syntax

new_zone_group AD_group_UPN

Abbreviation

newzg

Options

This command takes no options.

Arguments

This command takes the following argument:

AD_group_UPN	string	Required. Specifies the user principal name (UPN) of an Active Directory group.

Return Value

This command returns nothing if it runs successfully.

Examples

new_zone_group poweradmins@acme.com

This example creates a new zone group named poweradmins@acme.com in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- get_zone_groups returns a Tcl list of the Active Directory names of all zone groups in the current zone.
- list_zone_groups lists to stdout the zone groups in the current zone.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the current zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the current zone group.

new_zone_user

Use the new_zone_user command to create a new zone user in the current zone based on an existing Active Directory user. If the command is successful, the new zone user becomes the currently selected zone user stored in memory.

The new_zone_user command does not set any field values or save the new zone user to Active Directory. Before you can save the new zone user, you must first set at least one field value using the set_zone_user_field command. You can then save the zone user profile using the save_zone_user command.

Note: If the currently selected zone is a classic zone, you must set all fields for the new zone user before saving the user profile.

If you don't save a new zone user, it will disappear when you select another zone user or end the ADEdit session.

You can create more than one zone user within a zone based on a single Active Directory user. The first zone user you create uses the Active Directory user's user principal name (UPN), for example, martin.moore@acme.com. Any other zone users you create for the same Active Directory user must use aliases. An alias is the Active Directory user's UPN with +n appended where n is a positive integer that is unique for this Active Directory user in this zone. For example, martin.moore@acme.com+1 is an alias, as is martin.moore@acme.com+5. Alias integers need not be consecutive or in order. (Note that SFU zones do not support user aliases.)

The new_zone_user command requires you to specify Active Directory user name. The command will search for the user first by the supplied UPN in the specified domain, then by the **sAMAccountname** in the specified domain, then by the supplied UPN in any bound domain. If the user name cannot be found, the new zone user is not created.

Zone Type

Classic and hierarchical

Syntax

```
new_zone_user AD_user_UPN
```

Abbreviation

```
newzu
```

Options

This command takes no options.

Arguments

This command takes the following argument:

AD_user_UPN	string	Required. Specifies the user principal name (UPN) of an Active Directory user. If you are specifying an alias, append the UPN with "+" followed by a positive integer that is unique for this user and the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

```
new_zone_user adam.avery@acme.com
```

This example creates a new zone user based on the Active Directory user adam.avery@acme.com in the current zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone user:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone.
- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.

- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

pop

Use the pop command to retrieve a previously-stored context of bindings and selected objects from the top of the context stack. This command replaces the current ADEdit context with the retrieved context. Popping a context from the context stack removes the context from the stack.

This command is useful for Tcl scripts that use subroutines. A push can save the context before it's altered in the subroutine; a pop can return the saved context when the subroutine returns.

Zone Type

Not applicable

Syntax

```
pop
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully. If the stack is empty, it returns a message stating so.

Examples

```
pop
```

This example retrieves the context from the top of the context stack and uses it as the current ADEdit context.

Related Commands

The following commands perform actions related to this command:

- show returns the current context of ADEdit, including its bound domains and its currently selected objects.
- push saves the current ADEdit context to the ADEdit context stack.

principal_from_sid

Use the principal_from_sid command look up the security principal for a specified security identifier (SID) in Active Directory. If the security identifier is found, the command returns the Active Directory name of the principal.

Zone Type

Not applicable

Syntax

```
principal_from_sid [-upn] sid
```

Abbreviation

pfs

Options

This command takes the following option:

-upn	Returns the user names in user principal name (UPN) format, not the default sAMAccount@domain format.

Arguments

This command takes the following argument:

sid	string	Required. Specifies the security identifier of an Active Directory security principal.

Return Value

This command returns the Active Directory name of the principal if it finds a principal. If it does not find a principal, it returns a message stating so.

Examples

```
principal_from_sid S-1-5-21-2076040321-3326545908-468068287-1159
```

This example returns the principal name: oracle_machines@acme.com

Related Commands

The following commands perform actions related to this command:

- principal_to_dn searches Active Directory for a user principal name (UPN) and, if found, returns the corresponding distinguished name (DN).
- dn_to_principal searches Active Directory for a distinguished name (DN) and, if found, returns the corresponding user principal name (UPN).

principal_to_dn

Use the principal_to_dn command to search Active Directory for the specified user principal name (UPN) of a security principal (user, machine, or group). If a security principal is found for the specified UPN, the command returns the distinguished name (DN) of the principal.

Zone Type

Not applicable

Syntax

```
principal_to_dn principal_upn
```

Abbreviation

ptd

Options

This command takes no options.

Arguments

This command takes the following argument:

principal_upn	string	Required. Specifies the user principal name (UPN) of a security principal.

Return Value

This command returns a distinguished name. If the command doesn't find the specified security principal in Active Directory, it presents a message that it didn't find the principal.

Examples

principal_to_dn brenda.butler@acme.com

This example returns the distinguished name for the specified UPN:

cn=brenda butler,cn=users,dc=acme,dc=com

Related Commands

The following commands perform actions related to this command:

- dn_from_domain converts a domain's dotted name to a distinguished name.
- get_parent_dn returns the parent of an LDAP path as a distinguished name.
- get_rdn returns the relative distinguished name of an LDAP path.
- dn_to_principal searches Active Directory for a distinguished name, and, if found, returns the corresponding user principal name (UPN).
- principal_from_sid searches Active Directory for a security identifier and returns the security principal associated with the security identifier.

principal_to_id

Use the principal_to_id command to search Active Directory for the specified user principal name (UPN) of a user or group security principal. If a security principal is found for the specified UPN, the command returns the numeric identifier for the principal.

Zone Type

Not applicable

Syntax

principal_to_id [-apple] upn

Abbreviation

pti

Options

This command takes the following option:

-apple	Specifies that you want to use the Apple scheme for generating the UID or GID for the specified user or group principal. If you don't specify this option, the UID or GID returned is based on the Delinea Auto Zone scheme.

Arguments

This command takes the following argument:

upn	string	Required. Specifies the user principal name (UPN) of a user or group security principal.

Return Value

This command returns a unique UID or GID based on either the Apple methodology or the Delinea Auto Zone methodology for generating numeric identifiers. If the user or group principal is not found in Active Directory, the command returns an error message indicating that it didn't find the principal.

Examples

```
principal_to_id -apple brenda.butler@acme.com
```

This example returns the UID for the specified user generated using the Apple scheme:

```
1983765448
```

Related Commands

The following commands perform actions related to this command:

- guid_to_id accepts a globally unique identifier (GUID) for a user or group and returns a UID or GID generated using the Apple scheme.
- principal_from_sid searches Active Directory for a security identifier and returns the security principal associated with the security identifier.

push

Use the push command to save the current ADEdit context—its bindings and selected objects in memory—to a context stack. This command leaves the current context in place, so all current bindings and selected objects remain in effect in ADEdit after the push.

This command is useful for Tcl scripts that use subroutines. You can use the push command to save the context before it's altered in the subroutine. You can then use the pop command to retrieve the saved context when the subroutine returns.

Zone Type

Not applicable

Syntax

```
push
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing.

Examples

```
push
```

The example saves the current ADEdit context.

Related Commands

The following commands perform actions related to this command:

- show returns the current context of ADEdit, including its bound domains and currently selected objects.
- pop restores the context from the top of the ADEdit context stack to ADEdit.

quit

Use the quit command to quit ADEdit and return to the shell from which ADEdit was launched. You can also end an interactive ADEdit session by pressing Ctrl-D or entering exit.

> **Note**: If you enter the exit command, understand that it will terminate the session immediately without performing a commit operation.

Zone Type

Not applicable

Syntax

```
quit
```

Abbreviation

```
q
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing.

Examples

```
quit
```

This example ends an ADEdit session.

Related Commands

None.

remove_command_from_role

Use the remove_command_from_role command to remove a UNIX command from the currently selected role stored in memory.

The remove_command_from_role command does not change the role as it is stored in Active Directory. You must save the role before the removed command takes effect in Active Directory. If you select another role or quit ADEdit before saving the role, any UNIX commands you have removed since the last save won't take effect.

You can only use the remove_command_from_role command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

remove_command_from_role command[/zonename]

Abbreviation

rcfr

Options

This command takes no options.

Arguments

This command takes the following argument:

command[*/zonename*]	string	Required. Specifies the name of a UNIX command to remove from the currently selected role. If the UNIX command that you want to remove is defined in the current zone, the *zonename* argument is optional. If the UNIX command right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific command to remove.

Return Value

This command returns nothing if it runs successfully.

Examples

remove_command_from_role basicshell/global

This example removes the UNIX command named basicshell, which is defined in the global zone, from the currently selected role.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_roleadds a UNIX command to the current role.
- add_pamapp_to_role adds a PAM application to the current role. 'delete_role" deletes the selected role from Active Directory and from memory.
- get_role_apps returns a Tcl list of the PAM applications associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the current role.
- remove_pamapp_from_role removes a PAM application from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the current role.

remove_object_value

Use the remove_object_value command to remove a value from a multi-valued attribute of a specified Active Directory object. This command only affects the specified attribute for specified object in Active Directory. The command does not change the currently selected Active Directory object in memory, if there is one.

If the field or value to be removed isn't valid, Active Directory will report an error and remove_object_value won't remove the value.

This command is useful for fields that may be very large—members of a group, for example.

Zone Type

Not applicable

Syntax

```
remove_object_value dn field value
```

Abbreviation

rov

Options

This command takes no options.

Arguments

This command takes the following arguments:

dn	string	Required. Specifies the distinguished name (DN) of the Active Directory object from which to remove a value.
field	string	Required. Specifies the name of a multi-valued attribute in the currently selected Active Directory object from which to remove the value. This argument can be any field that is valid for the type of the Active Directory object you have specified using the *dn* argument. For example, if the Active Directory object specified is a computer object, the *field* argument might be operatingSystem.
value		Required. Specifies the value to remove from the field. The data type of the *value* depends on the *field* you specify.

Return Value

This command returns nothing if it runs successfully.

Examples

```
remove_object_value cn=groups,dc=acme,dc=com users adam.avery
```

This example removes the value adam.avery from the users field of the groups object in Active Directory.

Related Commands

The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects matching the search criteria.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object and its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_field sets a field value in the currently selected Active Directory object.

remove_pamapp_from_role

Use the remove_pamapp_from_role command to remove a PAM application access right from the currently selected role stored in memory.

The remove_pamapp_from_role command does not change the role as it is stored Active Directory. To remove the PAM application right from the role stored in Active Directory, you must save your changes using the save_role command. If you select another role or quit ADEdit before saving the role, any PAM applications you've removed since the last save won't take effect.

You can only use the remove_pamapp_from_role command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
remove_pamapp_from_role app[/zonename]
```

Abbreviation

```
rpamfr
```

Options

This command takes no options.

Arguments

This command takes the following argument:

app[*/zonename*]	string	Required. Specifies the name of a PAM application right to remove from the currently selected role. If the PAM application right that you want to remove is defined in the current zone, the *zonename* argument is optional. If the PAM application right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific PAM application right to remove.

Return Value

This command returns nothing if it runs successfully.

Examples

```
remove_pamapp_from_role ftp-all
```

This example removes the PAM application right named ftp-all defined in the currently selected zone from the currently selected role.

To remove the PAM application right when it is defined in the seattle zone, you would include the zone name:

```
remove_pamapp_from_role ftp-all/seattle
```

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

*add_command_to_roleadds a UNIX command to the current role.

- add_pamapp_to_role adds a PAM application to the current role. 'delete_role" deletes the selected role from Active Directory and from memory. *get_role_apps returns a Tcl list of the PAM applications associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- list_role_rights returns a list of all UNIX commands and PAM applications associated with the current role. *remove_command_from_roleremoves a UNIX command from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the current role.

remove_sd_ace

Use the remove_sd_ace command to remove an access control entry (ACE) in ACE string form from a security descriptor (SD) in SDDL (security descriptor description language) form.

The command looks for the supplied ACE string within the supplied SDDL string. If the command finds the ACE string, it removes it from the SDDL string and returns the SDDL string.

Zone Type

Not applicable

Syntax

remove_sd_ace sddl_string ace_string

Abbreviation

rsa

Options

This command takes no options.

Arguments

This command takes the following arguments:

sddl_string	string	Required. Specifies a security descriptor in SDDL format.
ace_string	string	Required. Specifies an access control entry in ACE string form, which is always enclosed in parentheses.

Return Value

This command returns a modified security descriptor in SDDL format if it runs successfully.

Examples

This example removes the first ACE string from an SDDL. The ACE string to remove is at the end of the command

(A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY):

remove_sd_ace O:DAG:DAD:AI (A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY) (A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA) (OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO) (A;;RCLCRPLO;;;AU) (OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015e5f28;;AO) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RCLCRPLO;;bf967a9c-0de6-11d0-a285-

00aa003049e2;RU) (OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS) (A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA) (A;CIID;LC;;;RU)(A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA) (A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY)

The command returns the SDDL string without the first ACE string:

O:DAG:DAD:AI (A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA) (OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO) (A;;RCLCRPLO;;;AU) (OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015e5f28;;AO) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RCLCRPLO;;bf967a9c-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS) (A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA) (A;CIID;LC;;;RU) (A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA) (A;;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;SY)

Related Commands

The following commands enable you to work with security descriptor strings:

- add_sd_ace adds an access control entry to a security descriptor.
- explain_sd converts an SD in SDDL format to a human-readable form.
- set_sd_owner sets the owner of a security descriptor.

rename_object

Use the rename_object command to rename the selected object. You can replace only the first relative distinguished name in the selected object. You do not need to save the object after you change the name.

Zone Type

Not applicable

Syntax

rename_object name

Abbreviation

rno

Options

This command takes no options.

Arguments

This command takes the following argument:

name	string	Required. Specifies the replacement relative distinguished name for the first relative distinguished name in the selected object.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example selects the user object Lois Lane and changes her name to LoisLane:

select_object “cn=Lois Lane,cn=Users,dc=demo,dc=test” rename_object LoisLane

The following example selects the organizational unit UnixServers an renames it to UNIX Servers:

```
select_object“ou=UnixServers,ou=Acme,dc=demo,dc=test” rno “UNIX Servers”
```

In both examples, quotes are required to preserve spaces in object names.

Related Commands

The following command performs actions related to this command:

- select_object selects the object you want to rename.

save_dz_command

Use the save_dz_command command to save the currently selected UNIX command stored in memory to Active Directory. You must save a UNIX command for any changes you make using ADEdit to take effect in Active Directory. If you select another UNIX command or end the ADEdit session before saving the currently selected UNIX command, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
save_dz_command
```

Abbreviation

```
svdzc
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
save_dz_command
```

This example saves the currently selected UNIX command to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- get_dzc_field reads a field value from the currently selected command.
- set_dzc_field sets a field value in the currently selected command.

save_local_group_profile

Use the save_local_group_profile command to save the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.

Whenever you execute the new_local_group_profile or select_local_group_profile command, the group continues to be selected until you execute the save_local_group_profile command.

You can save a group object before the group profile is complete. However, the group profile is not added to /etc/group on each UNIX and Linux computer in the zone until the group profile is complete and the profileflag field is set to 1 (enabled). See new_local_group_profile for details about which attributes a group profile must have to be considered complete.

Zone Type

Hierarchical only.

Syntax

save_local_group_profile

Abbreviation

svlgp

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example saves the currently selected object for the local UNIX or Linux group in the currently selected zone.

save_local_group_profile``

For example, earlier you might have executed the following command to select the marketing group object so that you could edit its profile fields:

select_local_group_profile marketing

Executing the following command would save any changes you had made to the marketing group object:

save_local_group_profile

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.

- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

save_local_user_profile

Use the save_local_user_profile command to save the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.

Whenever you execute the new_local_user_profile or select_local_user_profile command, the user continues to be selected until you execute the save_local_user_profile command.

You can save a user object before the user profile is complete. However, the user profile is not added to /etc/passwd on each UNIX and Linux computer in the zone until the user profile is complete, the profileflag field is set to 1 (enabled) or 2 (disabled), and the user is assigned a visible role such as local listed. See new_local_user_profile for details about which attributes a user profile must have to be considered complete.

Zone Type

Hierarchical only.

Syntax

```
save_local_user_profile
```

Abbreviation

```
svlup
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

T`his command returns nothing if it runs successfully.

Examples

The following example saves the currently selected object for the local UNIX or Linux user in the currently selected zone.

```
save_local_user_profile
```

For example, earlier you might have executed the following command to select the object for UNIX user anton.splieth so that you could edit its profile fields:

```
select_local_user_profile anton.splieth
```

Executing the following command would save any changes you had made to the user object for anton.splieth:

```
save_local_user_profile
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

save_nis_map

Use the save_nis_map command to save the currently selected NIS map stored in memory to Active Directory. You must save the NIS map for any changes you make using ADEdit to take effect in Active Directory. If you select another NIS map or end the ADEdit session before saving the currently selected NIS map, your changes will be lost.

Zone Type

Not applicable

Syntax

save_nis_map

Abbreviation

svnm

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_nis_map

This example saves the currently selected NIS map to Active Directory.

Related Commands

Before you use this command, you must have a currently selected NIS map stored in memory. The following commands enable you to view and select a NIS map:

- get_nis_maps returns a Tcl list of NIS maps in the current zone.
- list_nis_maps lists to stdout the NIS maps in the current zone.

- new_nis_map creates a new NIS map and stores it in memory.
- select_nis_map retrieves a NIS map from Active Directory and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

*add_map_entryOr add_map_entry_with_comment adds a map entry to the currently selected NIS map.

- delete_map_entry removes an entry from the currently selected NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the currently selected NIS map.
- get_nis_map_field reads a field value from the currently selected NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout the map entries in the currently selected NIS map.

save_object

Use the save_object command to save the currently selected Active Directory object stored in memory to Active Directory. You must save the Active Directory object for any changes you make using ADEdit to take effect in Active Directory. If you select another Active Directory object or end the ADEdit session before saving the currently selected object, your changes will be lost.

If an object has invalid attributes or values or is the wrong class for the container where it's being saved, Active Directory will report an error and the object will not be saved.

Zone Type

Not applicable

Syntax

save_object

Abbreviation

svo

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_object

This example saves the currently selected Active Directory object to Active Directory.

Related Commands

The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects matching the specified search criteria.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object and its attributes from Active Directory and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or

save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- set_object_field sets a field value in the currently selected Active Directory object.

save_pam_app

Use the save_pam_app command to save the currently selected PAM application access right stored in memory to Active Directory. You must save the PAM application right for any changes you make using ADEdit to take effect in Active Directory. If you select another PAM application right or end the ADEdit session before saving the currently selected PAM application right, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

save_pam_app

Abbreviation

svpam

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_pam_app

This example saves the currently selected PAM application to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a PAM application object:

- get_pam_apps returns a Tcl list of PAM applications in the current zone.
- list_pam_apps lists to stdout the PAM application rights in the current zone.
- new_pam_app creates a new PAM application right and stores it in memory.
- select_pam_app retrieves a PAM application right from Active Directory and stores it in memory.

After you have a PAM application right stored in memory, you can use the following commands to work with that PAM application:

- delete_pam_app deletes the selected PAM application from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application.
- set_pam_field sets a field value in the currently selected PAM application.

save_role

Use the save_role command to save the currently selected role stored in memory to Active Directory. You must save the role for any changes you make using ADEdit to take effect in Active Directory. If you select another role or end the ADEdit session before saving the currently selected role, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

save_role

Abbreviation

svr

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_role

This example saves the currently selected role to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select roles:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_roleadds a UNIX command to the current role.
- add_pamapp_to_role adds a PAM application right to the current role.
- delete role deletes the selected role from Active Directory and from memory. *get_role_apps returns a Tcl list of the PAM application rights associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the current role.
- list_role_rights returns a list of all UNIX commands and PAM application rights associated with the current role. *remove_command_from_roleremoves a UNIX command from the current role.
- remove_pamapp_from_role removes a PAM application right from the current role.
- set_role_field sets a field value in the current role.

save_role_assignment

Use the save_role_assignment command to save the currently selected role assignment stored in memory to Active Directory. You must save the role assignment for any changes you make using ADEdit to take effect in Active Directory. If you select another role assignment or end the ADEdit session before saving the currently selected role assignment, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
save_role_assignment
```

Abbreviation

```
svra
```

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

```
save_role_assignment
```

This example saves the currently selected role assignment to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select role assignment to work with:

- `get_role_assignments` returns a Tcl list of role assignments in the current zone.
- `list_role_assignments` lists to `stdout` the role assignments in the current zone.
- `new_role_assignment` creates a new role assignment and stores it in memory.
- `select_role_assignment` retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment's attributes, delete the role assignment, or save information for the role assignment:

- `delete_role_assignment` deletes the selected role assignment from Active Directory and from memory.
- `get_role_assignment_field` reads a field value from the current role assignment.
- `save_role_assignment` saves the selected role assignment with its current settings to Active Directory.
- `set_role_assignment_field` sets a field value in the current role assignment.
- `write_role_assignment` saves the selected role assignment to a file.

save_rs_command

Use the `save_rs_command` command to save the currently selected restricted shell command that is stored in memory to Active Directory. You must save the restricted shell command for any changes you make using ADEdit to take effect in Active Directory. If you select another restricted shell command or end the ADEdit session before saving the currently selected restricted shell command, your changes will be lost.

Zone Type

Classic only

Syntax

```
save_rs_command``
```

Abbreviation

svrsc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_rs_command

This example saves the currently selected RSC to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- set_rsc_field sets a field value in the currently selected command.

save_rs_env

Use the save_rs_env command to save the currently selected restricted shell environment that is stored in memory to Active Directory. You must save the selected restricted shell environment for any changes you make using ADEdit to take effect in Active Directory. If you select another restricted shell environment or end the ADEdit session before saving the currently selected restricted shell environment, your changes will be lost.

Zone Type

Classic only

Syntax

save_rs_env

Abbreviation

svrse

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_rs_env

This command saves the currently selected restricted shell environment to Active Directory.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- set_rse_field sets a field value in the current restricted shell environment.

save_zone

Use the save_zone command to save the currently selected zone stored in memory to Active Directory. You must save the selected zone for any changes you make using ADEdit to take effect in Active Directory. If you select another zone or end the ADEdit session before saving the currently selected zone, your changes will be lost.

This command only saves fields that are properties in the currently selected zone. The command does not save any users or groups added to a zone. You must save users and groups individually using the save_zone_user and save_zone_group commands.

Zone Type

Classic and hierarchical

Syntax

save_zone

Abbreviation

svz

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_zone

This example saves the currently selected zone or computer role to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone to work with:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.
- select_zone retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- set_zone_field sets a field value in the currently selected zone.

save_zone_computer

Use the save_zone_computer command to save the currently selected zone computer stored in memory to Active Directory. You must set at least one field value before you can save a zone computer. In classic zones, you must set all field values before you can save a zone computer.

You must save the selected zone computer for any changes you make using ADEdit to take effect in Active Directory. If you select another zone computer or end the ADEdit session before saving the currently selected zone computer, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

save_zone_computer

Abbreviation

svzc

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_zone_computer

This example saves the currently selected zone computer to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the zone computers:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

save_zone_group

Use the save_zone_group command to save the currently selected zone group stored in memory to Active Directory. You must set at least one field value before you can save a zone group. In classic zones, you must set all field values before you can save a zone group.

You must save the selected zone group for any changes you make using ADEdit to take effect in Active Directory. If you select another zone group or end the ADEdit session before saving the currently selected zone group, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

save_zone_group

Abbreviation

svzg

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_zone_group

This example saves the currently selected zone group to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- get_zone_groups returns a Tcl list of the Active Directory names of all zone groups in the current zone.
- list_zone_groups lists to stdout the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the currently selected zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

save_zone_user

Use the save_zone_user command to save the currently selected zone user stored in memory to Active Directory. You must set at least one field value before you can save a zone user. In classic zones, you must set all field values before you can save a zone user.

You must save the selected zone user for any changes you make using ADEdit to take effect in Active Directory. If you select another zone user or end the ADEdit session before saving the currently selected zone user, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

save_zone_user

Abbreviation

svzu

Options

This command takes no options.

Arguments

This command takes no arguments.

Return Value

This command returns nothing if it runs successfully.

Examples

save_zone_user

This example saves the currently selected zone user to Active Directory.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone user:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone.
- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

select_dz_command

Use the select_dz_command command to retrieve a UNIX command in the currently selected zone from Active Directory. This command stores the selected UNIX command in memory, and makes it the currently selected UNIX command for subsequent ADEdit commands. The UNIX command remains selected until you select another UNIX command or zone, delete the UNIX command, or end the ADEdit session.

If you use ADEdit commands such as set_dzc_field to change settings for the selected UNIX command, you must save the selected UNIX command using the save_dz_command command for your changes to take effect in Active Directory. If you select another UNIX command or end the ADEdit session before saving the currently selected UNIX command, your changes will be lost.

You can only use the select_dz_command command to select UNIX commands if the currently selected zone is a classic4 or hierarchical zone. The command does not work for other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
select_dz_command command
```

Abbreviation

sldzc

Options

This command takes no options.

Arguments

This command takes the following arguments:

command	string	Required. Specifies the name of the UNIX command to select.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_dz_command account_manager
```

This example looks for the UNIX command named "account_manager" in the current zone and, if found, selects it as the current UNIX command.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command to work with:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- get_dzc_field reads a field value from the currently selected command.
- save_dz_command saves the selected command with its current settings to Active Directory.

- set_dzc_field sets a field value in the currently selected command.

select_local_group_profile

Use the select_local_group_profile command to select a local UNIX or Linux group object for viewing or editing. The group that you specify remains selected until you execute the save_local_group_profile command.

You typically use select_local_group_profile to select a group profile before you execute get_local_group_profile_field or set_local_group_profile_field to view or edit profile information.

Zone Type

Hierarchical only.

Syntax

```
select_local_group_profile group_name
```

Abbreviation

sllgp

Options

This command takes no options.

Arguments

This command takes the following argument:

group_name	string	Required. Specifies the UNIX name of the local group to select.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example selects the object for the local UNIX or Linux group marketing.

```
select_local_group_profile marketing
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.

- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

select_local_user_profile

Use the select_local_user_profile command to select a local UNIX or Linux user object for viewing or editing. The user that you specify remains selected until you execute the save_local_user_profile command.

You typically use select_local_user_profile to select a user profile before you execute get_local_user_profile_field or set_local_user_profile_field to view or edit profile information.

Zone Type

Hierarchical only.

Syntax

```
select_local_user_profile user_name
```

Abbreviation

sllup

Options

This command takes no options.

Arguments

This command takes the following argument:

user_name	string	Required. Specifies the UNIX name of the local user to select.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example selects the object for the local UNIX or Linux user anton.splieth.

```
select_local_user_profile anton.splieth
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.

- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

select_nis_map

Use the select_nis_map command to retrieve a NIS map in the currently selected zone from Active Directory. This command stores the NIS map in memory, and makes it the currently selected NIS map for subsequent ADEdit commands. The NIS map remains selected until you select another NIS map or zone, delete the NIS map, or end the ADEdit session.

If you use ADEdit commands such asadd_map_entryto change settings for the selected NIS map, you must save the selected NIS map using the save_nis_map command for your changes to take effect in Active Directory. If you select another NIS map or end the ADEdit session before saving the currently selected NIS map, your changes will be lost.

Zone Type

Not applicable

Syntax

```
select_nis_map map
```

Abbreviation

```
slnm
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

map	string	Required. Specifies the name of the NIS map to retrieve from Active Directory.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_nis_map Printers
```

This example looks for the NIS map named “Printers” in the current zone and, if found, selects it as the current NIS map.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select NIS maps:

- get_nis_maps returns a Tcl list of NIS maps in the current zone.
- list_nis_maps returns a list to stdout of all NIS maps in the current zone.
- new_nis_map creates a new NIS map and stores it in memory.

After you have a NIS map stored in memory, you can use the following commands to work with that map:

- add_map_entryor add_map_entry_with_comment adds an entry to the current NIS map stored in memory.
- delete_map_entry removes an entry from the current NIS map.
- delete_nis_map deletes the selected NIS map from Active Directory and from memory.
- get_nis_map or get_nis_map_with_comment returns a Tcl list of the map entries in the current NIS map.
- get_nis_map_field reads a field value from the current NIS map.
- list_nis_map or list_nis_map_with_comment lists to stdout the map entries in the current NIS map.
- save_nis_map saves the selected NIS map with its current entries to Active Directory.

select_object

Use the select_object command to retrieve the specified Active Directory object and its attributes from Active Directory. This command stores the object in memory and makes it the currently selected Active Directory object. You can use options to retrieve the rootDSE of the object or to list specific attributes to retrieve for the object.

Zone Type

Not applicable

Syntax

```
select_object [-rootdse] [-attrs a1[,a2,...]] dn
```

Abbreviation

slo

Options

This command takes the following options:

-rootdse	Returns the rootDSE of the specified object instead of the object.
-attrs *a1*[,a2,...]	Specifies the attributes to retrieve and store in memory. If you use this option, only the attributes you name (*a1, a2, a3,* and so on) are retrieved. This option is useful if you want to limit the number of attributes returned or want to return attributes not normally returned by Active Directory. If you do not use this option, ADEdit retrieves the attributes normally returned by Active Directory for the selected object type.

Arguments

This command takes the following argument:

dn	DN	Required. Specifies the distinguished name (DN) of an Active Directory object.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_object “cn=users,dc=acme,dc=com”
```

This example returns the container object cn=users,dc=acme,dc=com and its attributes, and stores it in memory as the currently selected Active Directory object.

Related Commands

The following commands enable you to view and select the object to work with:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects matching the specified search criteria.
- new_object creates a new Active Directory object and stores it in memory.

After you have an Active Directory object stored in memory, you can use the following commands to work with that object's attributes, delete the object, or save information for the object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.
- set_object_field sets a field value in the currently selected Active Directory object.

select_pam_app

Use the select_pam_app command to retrieve a PAM application access right in the currently selected zone from Active Directory. This command stores the PAM application right in memory, and makes it the currently selected PAM application right for subsequent ADEdit commands. The PAM application right remains selected until you select another PAM application right or zone, delete the PAM application right, or end the ADEdit session.

If you use ADEdit commands such as set_pam_field to change settings for the selected PAM application right, you must save the selected PAM application right using the save_pam_app command for your changes to take effect in Active Directory. If you select another PAM application right or end the ADEdit session before saving the currently selected PAM application right, your changes will be lost.

You can only use the select_pam_app command to select PAM applications if the currently selected zone is a classic4 or hierarchical zone. The command does not work for other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
select_pam_app name[/zonename]
```

Abbreviation

slpam

Options

This command takes no options.

Arguments

This command takes the following argument:

name[*/zonename*]	string	Required. Specifies the name of the PAM application right to select. If the PAM application right that you want to select is defined in the current zone, the *zonename* argument is optional. If the PAM application right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific PAM application right to select.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example retrieves the PAM application right named sftp in the current zone and makes it the currently selected PAM application right:

```
select_pam_app sftp
```

The following example retrieves the PAM application right named sftp defined in the chicago zone and makes it the currently selected PAM application right:

```
select_pam_app sftp/chicago
```

The definition for the PAM application right named sftp might be the same in both zones, but it is not required to be. Specifying the zone ensures you get the definition you expect.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. After you have a zone stored in memory, you can use the following commands to view and select the PAM application to work with:

- get_pam_apps returns a Tcl list of PAM application rights in the current zone.
- list_pam_apps lists to stdout the PAM application rights in the current zone.
- new_pam_app creates a new PAM application right and stores it in memory.
- select_pam_app retrieves a PAM application right from Active Directory and stores it in memory

After you have a PAM application stored in memory, you can use the following commands to work with that PAM application's attributes, delete the PAM application, or save information for the PAM application:

- delete_pam_app deletes the selected PAM application right from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application right.
- save_pam_app saves the selected PAM application right with its current settings to Active Directory.
- set_pam_field sets a field value in the currently selected PAM application right.

select_role

Use the select_role command to retrieve a role in the currently selected zone from Active Directory. This command stores the role in memory, and makes it the currently selected role for subsequent ADEdit commands. The role remains selected until you select another role or zone, delete the role, or end the ADEdit session.

If you use ADEdit commands such as set_role_field to change settings for the selected role, you must save the selected role using the save_role command for your changes to take effect in Active Directory. If you select another role or end the ADEdit session before saving the currently selected role, your changes will be lost.

You can only use the select_role command to select roles if the currently selected zone is a classic4 or hierarchical zone. The command does not work for other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
select_role role
```

Abbreviation

```
slr
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

role	string	Required. Specifies the name of the role to select.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_role servicerep
```

This example retrieves the role definition named servicerep in the current zone and makes it as the currently selected role.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a role:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

*add_command_to_roleadds a UNIX command right to the current role.

- add_pamapp_to_role adds a PAM application right to the current role.
- delete-role deletes the selected role from Active Directory and from memory. *get_role_apps returns a Tcl list of the PAM application rights associated with the current role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the current role.
- list_role_rights returns a list of all UNIX command and PAM application rights associated with the current role. *remove_command_from_roleremoves a UNIX command right from the current role.
- remove_pamapp_from_role removes a PAM application right from the current role.
- save_role saves the selected role with its current settings to Active Directory.
- set_role_field sets a field value in the current role.

select_role_assignment

Use the select_role_assignment command to retrieve a role assignment in the currently selected zone from Active Directory. This command stores the role assignment in memory, and makes it the currently selected role assignment for subsequent ADEdit commands. The role assignment remains selected until you select another role assignment or zone, delete the role assignment, or end the ADEdit session.

If you use ADEdit commands such as set_role_assignment_field to change settings for the selected role assignment, you must save the selected role assignment using the save_role_assignment command for your changes to take effect in Active Directory. If you select another role assignment or end the ADEdit session before saving the currently selected role assignment, your changes will be lost.

You can only use the select_role_assignment command to select role assignments if the currently selected zone is a classic4 or hierarchical zone. The command does not work for other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
select_role_assignment principal/role[/zone]
```

Abbreviation

slra

Options

This command takes no options.

Arguments

This command takes the following argument:

principal/role[/zone]	string	Required. Specifies the user principal name (UPN) of the user or group to whom the role is assigned, followed by a slash (/) and the name of the role to assign to the principal. The *zone* argument is optional if the role is defined in the currently selected zone. If the role is defined in a zone other than the currently selected zone, the */zone* argument is required.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_role_assignment poweradmins@acme.com/root/global
```

This example retrieves the role assignment that assigns the role named root, as defined in the global zone, to the principal named poweradmins@acme.com. The principal is a group.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a role assignment:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- list_role_assignments lists to stdout the role assignments in the current zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the currently selected role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- set_role_assignment_field sets a field value in the currently selected role assignment.
- write_role_assignment saves the selected role assignment to a file.

select_rs_command

Use the select_rs_command command to retrieve a restricted shell command in the currently selected zone from Active Directory, store it in memory, and set it as the currently selected restricted shell command for other ADEdit commands. After you select the restricted shell command to work with, it remains selected until you select a different restricted shell command, change the currently selected zone, delete the restricted shell command, or end the ADEdit session.

If you use ADEdit commands such as set_rsc_field to change settings for the selected restricted shell command, you must save the restricted shell command using the save_rs_command command for your changes to take effect in Active Directory. If you select another restricted shell command or end the ADEdit session before saving the currently selected restricted shell command, your changes will be lost.

You can only use the select_rs_command if the currently selected zone is a classic zone.The command does not work in other types of zones.

Zone Type

Classic only

Syntax

select_rs_command rs_cmd

Abbreviation

slrsc

Options

This command takes no options.

Arguments

This command takes the following argument:

rs_cmd	string	Required. Specifies the name of the restricted shell command to select.

Return Value

This command returns nothing if it runs successfully.

Examples

select_rs_command rsc1

This command looks for the restricted shell command name rsc1 in the current zone. If rsc1 is found in the current zone, it becomes the currently selected context for subsequent commands.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell command:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.
- set_rsc_field sets a field value in the currently selected command.

select_rs_env

Use the select_rs_env command to retrieve a restricted shell environment in the currently selected zone from Active Directory, stores it in memory, and sets it to be the currently selected restricted shell environment for other ADEdit commands. The restricted shell environment remains selected until you select another restricted shell environment, change the currently selected zone, delete the restricted shell environment, or end the ADEdit session.

If you use ADEdit commands such as set_rse_field to change settings for the restricted shell environment, you must save the restricted shell environment using the save_rs_env command for your changes to take effect in Active Directory. If you select another restricted shell environment or end the ADEdit session before saving the currently selected restricted shell environment, your changes will be lost.

You can only use the select_rs_env command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

select_rs_env rse_name

Abbreviation

slrse

Options

This command takes no options.

Arguments

This command takes the following argument:

rse_name	string	Required. Specifies the name of the restricted shell environment to select.

Return Value

This command returns nothing if it runs successfully.

Examples

select_rs_env rse1

This command looks for the restricted shell environment named rse1 in the current zone. If rse1 is found in the current zone, it becomes the currently selected context for subsequent commands.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.
- set_rse_field sets a field value in the current restricted shell environment.

select_zone

Use the select_zone command to retrieve a zone from Active Directory, stores the zone in memory, and make that zone as the currently selected zone for subsequent ADEdit commands. The zone remains selected until you select another zone, delete the zone, or end the ADEdit session.

If you use ADEdit commands such as set_zone_field to change settings for the zone, you must save the zone using the save_zone command for your changes to take effect in Active Directory. If you select another zone or end the ADEdit session before saving the currently selected zone, your changes will be lost.

You should note that ADEdit treats *computer roles* and *computer-specific overrides* as special use-case zones. You can, therefore, use the select_zone command to retrieve a “computer role zone” or a “computer-specific zone” to work with as the currently selected zone. If you specify a zone that is a computer role zone or a computer-specific zone, subsequent ADEdit commands will treat the zone as a computer role or a computer-specific zone instead of a standard zone. You can only work with one zone at a time, regardless of type. Because some ADEdit commands behave differently in different types of

zones, you should verify the type of zone you are working with when you select a zone.

Zone Type

Classic and hierarchical

Syntax

```
select_zone [-nc] path
```

Abbreviation

slz

Options

This command takes the following option:

-nc	Requests a reread of the zone's fields from Active Directory. Use this option after you use the save_zone command to ensure you have the current Active Directory field values in memory. For example, after a save_zone command, the modifyTime field value is updated. If you do not then run select_zone -nc, a gzf modifyTime command returns the previous value.

Arguments

This command takes the following argument:

path	string	Required. Specifies the path to the selected zone or computer role. The path format depends on the type of zone selected: A tree, classic3, classic4, or SFU zone path consists of the zone's distinguished name. Enclose the path in braces or quotes to allow spaces in the distinguished name. A computer role path consists of the host zone's distinguished name followed by a slash (/) and the name of the computer zone. Enclose the path in braces or quotes to allow spaces in the distinguished name. A computer override path consists of the computer name followed by an ampersand (@) and the distinguished name of the host zone.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example selects a standard zone named cz1 in the Zones container in the UNIX organizational unit in the acme.com domain:

```
select_zone "CN=cz1,CN=Zones,OU=UNIX,DC=acme,DC=com”
```

The following example selects the computer role named LinuxComputers in the global zone in the Zones container in the UNIX organizational unit in the acme.com domain:

```
select_zone “CN=global,CN=Zones,OU=UNIX,DC=acme,DC=com/LinuxComputers”
```

The following example selects the computer-specific override zone named server1 in the global zone in the acme.com domain:

```
select_zone "server1.acme.com@CN=global,CN=Zones,OU=Acme,DC=acme,DC=com”
```

Related Commands

The following commands perform actions related to this command:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.
- set_zone_field sets a field value in the currently selected zone.

select_zone_computer

Use the select_zone_computer command to retrieve a zone computer in the currently selected zone from Active Directory, store it in memory, and make it the currently selected zone computer for subsequent ADEdit commands. The zone computer remains selected until you select another zone computer, delete the zone computer, or end the ADEdit session.

If you use ADEdit commands such as set_zone_computer_field to change settings for the zone computer, you must save the zone computer using the save_zone_computer command for your changes to take effect in Active Directory. If you select another zone computer or end the ADEdit session before saving the currently selected zone computer, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
select_zone_computer sAMAccountName$@domain
```

Abbreviation

slzc

Options

This command takes no options.

Arguments

This command takes the following argument:

sAMAccountName	string	Required. Specifies the Active Directory computer's sAMAccountName followed by $@ and the computer's domain. You can look up the sAMAccountName for a computer in Active Directory Users and Computers or by running the get_zone_computers command.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_zone_computer sales2$@acme.com
```

This example looks for the zone computer named sales2 in the current zone and, if found, selects it as the current zone computer.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the zone computers:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

select_zone_group

Use the select_zone_group command to retrieve a zone group in the currently selected zone from Active Directory. The command stores the zone group in memory and makes it the currently selected zone group for subsequent ADEdit commands. The zone group remains selected until you select another zone group, delete the zone group, or end the ADEdit session.

If you use ADEdit commands such as set_zone_group_field to change settings for the zone group, you must save the zone group using the save_zone_group command for your changes to take effect in Active Directory. If you select another zone group or end the ADEdit session before saving the currently selected zone group, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
select_zone_group AD_group_UPN
```

Abbreviation

```
slzg
```

Options

This command takes no options.

Arguments

This command takes the following argument:

AD_group_UPN	string	Required. Specifies the user principal name (UPN) of a zone group in the currently selected zone.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_zone_group poweradmins@acme.com
```

This example looks for the group named poweradmins in the current zone and, if found, selects it as the current zone group.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- get_zone_groups returns a Tcl list of the Active Directory names of all zone groups in the current zone.

- list_zone_groups lists to stdout the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the currently selected zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.
- set_zone_group_field sets a field value in the currently selected zone group.

select_zone_user

Use the select_zone_user command to retrieve a zone user in the currently selected zone from Active Directory. This command stores the zone user in memory, and makes it the currently selected zone user for subsequent ADEdit commands. The zone user remains selected until you select another zone user, delete the zone user, or end the ADEdit session.

If you use ADEdit commands such as set_zone_user_field to change settings for the zone user, you must save the zone user using the save_zone_user command for your changes to take effect in Active Directory. If you select another zone user or end the ADEdit session before saving the currently selected zone user, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
select_zone_user user
```

Abbreviation

slzu

Options

This command takes no options.

Arguments

This command takes the following argument:

user	string	Required. Specifies the sAMAccountName@domain or user principal name (UPN) of a zone user in the currently selected zone. ADEdit resolves the user with the sAMAcccountName first, then the UPN. If the zone user is an orphan user—that is, the corresponding Active Directory user no longer exists—you must specify the user's security identifier (SID) instead.

Return Value

This command returns nothing if it runs successfully.

Examples

```
select_zone_user adam.avery@acme.com
```

This example looks for the Active Directory user adam.avery in the current zone and, if found, selects that user as the current zone user.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone user:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone.
- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.
- set_zone_user_field sets a field value in the currently selected zone user.

set_dzc_field

Use the set_dzc_field command to set the value for a specified field in the currently selected UNIX command stored in memory. The set_dzc_field command does *not* set a field value stored in Active Directory for the selected UNIX command.

If you change any fields, you must save the UNIX command using the save_dz_command command for your changes to take effect in Active Directory. If you select another UNIX command or end the ADEdit session before saving the currently selected UNIX command, your changes will be lost.

You can only use the set_dzc_field command to set UNIX command fields if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

When executing privileged commands on computers running Security-Enhanced Linux (SELinux), the security context contains additional information that is used to make access control decisions.

Zone Type

Classic and hierarchical

Syntax

set_dzc_field field value

Abbreviation

sdzcf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field you want to set. The possible values are: **description**: Text describing the UNIX command. **cmd**: The UNIX command string or strings. You can use wild cards or a regular expression. **path**: The path to the command's location. You can use wild cards or a regular expression. **form**: An integer that indicates whether the cmd and path strings use wild cards (0) or a regular expression (1). **dzdo_runas**: A list of users and groups that can run this command under dzdo (similar to sudo). Users can be listed by user name or UID. **dzsh_runas**: A list of users and groups that can run this command in a restricted shell environment (dzsh). Users can be listed by user name or UID. You cannot set this field value if the selected zone is a classic4 zone. **keep**: A comma-separated list of environment variables from the current user's environment to keep. **del**: A comma-separated list of environment variables from the current user's environment to delete. **add**: A comma-separated list of environment variables to add to the final set of environment variables. **pri**: An integer that specifies the command priority for the restricted shell command object.
		umask: An integer that defines who can execute the command. **flags**: An integer that specifies a combination of different properties for the command. **selinux_role**: Specifies the SELinux role to use when constructing a new security context for command

field (continued)	string	execution. **selinux_type**: Specifies the SELinux type to use when constructing a new security context for command execution. **digest**: Specifies the SHA-2 digest to verify the file checksum before command execution. Note that selinux_role and selinux_type are only supported on Red Hat Enterprise Linux systems and effective only on systems with SELinux enabled and joined to a hierarchical zone.
value		Required. Specifies the value to assign to the specified field. The data type depends on the field specified. In most cases, you can assign an empty string or null value (0) to unset a field value, depending on the data type of the field.

Setting the cmd and path field values

You can specify the cmd and path strings using wild cards (*, ?, and !), or as a regular expression. If you specify the cmd and path strings using wild cards, use an asterisk (*) to match zero or more characters, the question mark (?) to match exactly one character, or the exclamation mark (!) to negate matching of the specified string.

To set to the command path to the equivalent of the **Standard user path** option, you can set the value of the path field to USERPATH. To set to the path to the equivalent of the **Standard system path** option, set the value of the path field to SYSTEMPATH. To set to the path to the equivalent of the **System search path** option, set the value of the path field to SYSTEMSEARCHPATH.

For both the cmd and path fields, the form field controls whether the specified string is interpreted as a regular expression or as a string that includes wild cards.

Specifying the environment variables to use

You can use the keep, del, and add settings to control the environment variables used by the commands specified by the cmd string. The keep and del settings are mutually exclusive. The keep field only takes effect if the flag 16 is included in the setting for the flag field. The del field only takes effect if the flag 16 is not included in the setting for the flag field.

Any environment variables kept or deleted are in addition to the default set of the user's environment variables that are either retained or deleted. The default set of environment variables to keep is defined in the dzdo.env_keep configuration parameter in the centrifydc.conf file. The default set of environment variables to delete is defined in the dzdo.env_delete configuration parameter in the centrifydc.conf file. You can also add environment variables to the final set of environment variables resulting from the keep or del fields.

Specifying the command priority

You can use the pri field to specify the command priority when there are multiple matches for the UNIX commands specified by wild cards. If commands specified by this UNIX command object match commands specified by another UNIX command object, the UNIX command object with the higher command priority prevails. This field takes an integer value; the higher the number, the higher the priority.

Specifying the umask value

You can use the umask field to define who can execute the command. The umask field specifies a 3-digit octal value that defines read, write, or execute permission for owner, group, and other users. The left digit defines the owner execution rights, the middle digit defines the group execution rights, and the right digit defines other execution rights. Each digit is a combination of binary flags, one flag for each right as follows:

- 4 is read
- 2 is write
- 1 is execute

You add these values add together to define the rights available for each entity. For example, an umask value of 600 indicates read and write permission (4+2) for the owner, but no permissions for the group or other users. Similarly, an umask value of 740 indicates read, write, execute permissions (4+2+1) for the owner, read permissions for the group, but no permissions for other users.

Specifying command properties using the flags field

You can use the flags field to define a combination of binary flags, with one flag for each of the following properties:

1—Prevents nested command execution. If this flag value is not set, nested command execution is allowed.

2—Requires re-authentication using the login user's password.

4—Requires authentication using the run-as user's password.

8—Preserves group membership. If this flag value is not set, group membership is not preserved.

16—Resets environment variables for the command, deleting the variables specified in the dzdo.env_delete parameter and keeping the variables specified in the keep field. If this flag is not set, the command removes the unsafe environment variables specified in the dzdo.env_delete parameter along with any additional environment variables specified by the del field.

32—Requires multi-factor authentication to execute the command.

64—Prevents navigation up the path hierarchy when executing the command.

You add these values together to define the setting for the flags field. For example, a flags field value of 5 prevents nested command execution and requires authentication using the run-as user's password (1+4). You cannot set the 2 flag and the 4 flag or the 4 flag and the 32 flag simultaneously. If you don't set any of these flags, re-authentication is not required.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the current UNIX command dzdo_runas field to root:

```
set_dzc_field dzdo_runas root
```

The following example sets the UNIX command properties so that nested command execution is not allowed and authentication is required with the user's password:

```
sdzcf flags 3
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a UNIX command to work with:

- get_dz_commands returns a Tcl list of UNIX commands in the current zone.
- list_dz_commands lists to stdout the UNIX commands in the current zone.
- new_dz_command creates a new UNIX command and stores it in memory.
- select_dz_command retrieves a UNIX command from Active Directory and stores it in memory.

After you have a UNIX command stored in memory, you can use the following commands to work with that command:

- delete_dz_command deletes the selected command from Active Directory and from memory.
- get_dzc_field reads a field value from the currently selected command.
- save_dz_command saves the selected command with its current settings to Active Directory.

set_ldap_timeout

Use the set_ldap_timeout command to set the time-out interval used by LDAP commands. LDAP commands are ADEdit commands such as select_zone that perform read/write operations on Active Directory through a binding. The time-out value controls how long these commands will wait for a response before declaring a time-out and ceasing operation.

The default value is five minutes.

Zone Type

Not applicable

Syntax

```
set_ldap_timeout timeout_in_seconds
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following argument:

timeout_in_seconds	integer	Required. Specifies the number of seconds to wait for a response from Active Directory before ending an operation. The default value is 300 seconds (5 minutes).

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_ldap_timeout 120
```

This example sets the LDAP time-out interval to 120 seconds (2 minutes).

Related Commands

None.

set_local_group_profile_field

Use the set_local_group_profile_field command to set the value of the specified profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone. Before executing this command, you must create a new local group by executing the new_local_group_profile command, or select an existing local group by executing the select_local_group_profile command.

You can save a group object before the group profile is complete. However, the group profile is not added to /etc/group on each UNIX and Linux computer in the zone until the group profile is complete and the profileflag field is set to 1 (enabled). See new_local_group_profile for details about which fields (attributes) a group profile must have to be considered complete.

Zone Type

Hierarchical only.

Syntax

```
set_local_group_profile_field field_name value
```

Abbreviation

```
slgpf
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

field_name	string	Required. Specifies the local group profile field to set. The possible values are: **gid member profileflag** You can also specify AIX extended attributes as the field to set an extended attribute value for a group. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be set by specifying aix.admin as the field.
value		Required. The data type depends on the field being set. The possible values for each field are: **Any field**: Clear any field by entering a hyphen character (-). **gid**: A numeric group identifier. **member**: The UNIX name of a local user to add to the group. **profileflag**: 1 or 3.If set to 1, the group profile is enabled. If the group profile is complete and the profile flag is set to 1, the profile will be installed or updated in /etc/group at the next local account refresh interval. If set to 3, the group profile is removed from /etc/group at the next local account refresh interval.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the GID of the currently selected group to 20001.

```
set_local_group_profile_field gid 20001
```

The following example adds the UNIX user anton.splieth to the currently selected local group.

```
set_local_group_profile_field member anton.splieth
```

The following example sets the profile flag of the currently selected group to 1 (enabled), so that if the group profile is complete, the profile will be installed or updated in /etc/group at the next local account refresh interval.

```
set_local_group_profile_field profileflag 1
```

If the current group is on AIX, you can set group AIX extended attributes and values. For example, to identify the current group as an administrative group, you can set the admin extended attribute:

```
set_local_group_profile_field aix.admin true
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_user_profile_field sets the value of a field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.

set_local_user_profile_field

Use the set_local_user_profile_field command to set the value of the specified profile field for the currently selected local UNIX or Linux user that has a profile

defined in the current zone. Before executing this command, you must create a new local user by executing the new_local_user_profile command, or select an existing local user by executing the select_local_user_profile command.

You can save a user object before the user profile is complete. However, the user profile is not added to /etc/passwd on each UNIX and Linux computer in the zone until the user profile is complete, the profileflag field is set to 1 (enabled) or 2 (disabled), and the user is assigned a visible role such as local listed. See new_local_user_profile for details about which attributes a user profile must have to be considered complete.

Zone Type

Hierarchical only.

Syntax

```
set_local_user_profile_field field_name value
```

Abbreviation

slupf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field_name value	String	Required. Specifies the local user profile field to set. Fields and possible values are: **Any field**: Clear any field by entering a hyphen character (-). **uid**: The user's numeric identifier. **gid**: The user's primary group numeric identifier. **shell**: The local user's default shell on the local computer. Possible values are: /bin/bash, /bin/csh, /bin/ksh, /bin/sh, /bin/tcsh, %. **home**: The local user's default home directory on the local computer. **gecos**: General information about the local user account. **profileflag**: The value of the user's profile flag as set in the user object in the zone. For the user to be managed by the agent, the profile flag must be set to 1 , 2, or 3.If set to 1, the user profile is enabled. If the user profile is complete, the profile flag is set to 1, and the user is assigned a visible role, the profile will be installed or updated in /etc/passwd at the next local account refresh interval. If set to 2, the user profile is disabled. If the user profile is complete, the profile flag is set to 2, and the user is assigned a visible role, the profile will be installed or updated in /etc/passwd at the next local account refresh interval. However, the password field in /etc/passwd will be set to !!, and the user will not be able to log into the local computer. This state results in what is typically called a "locked account." If set to 3, the user profile is removed from /etc/passwd at the next local account refresh interval.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the UID of the currently selected user to 10001.

```
set_local_user_profile_field uid 10001
```

The following example sets the primary group ID for the currently selected user to 20001.

```
set_local_user_profile_field gid 20001
```

The following example sets the default shell for the currently selected user to /bin/csh:

```
set_local_user_profile_field shell /bin/csh
```

The following example sets the home directory for the currently selected user to /home.

```
set_local_user_profile_field home /home
```

The following example sets the profile flag of the currently selected user to 1 (enabled), so that if the user profile is complete and the user is assigned a visible role, the profile will be installed or updated in /etc/passwd at the next local account refresh interval.

```
set_local_user_profile_field profileflag 1
```

Related Commands

The following related ADEdit commands let you view and administer local UNIX and Linux users and groups that have profiles defined in the current zone:

- delete_local_group_profile deletes a local UNIX or Linux group that has a profile defined in the current zone.
- delete_local_user_profile deletes a local UNIX or Linux user that has a profile defined in the current zone.
- get_local_group_profile_field displays the value of a profile field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.
- get_local_groups_profile displays a TCL list of profiles for local groups that are defined in the current zone.
- get_local_user_profile_field displays the value of a profile field for the currently selected local UNIX or Linux user that has a profile defined in the current zone.
- get_local_users_profile displays a TCL list of profiles for local users that are defined in the current zone.
- list_local_groups_profile displays a list of local UNIX and Linux groups that have a profile defined in the current zone.
- list_local_users_profile displays a list of local UNIX and Linux users that have a profile defined in the current zone.
- new_local_group_profile creates an object for a local UNIX or Linux group in the currently selected zone.
- new_local_user_profile creates an object for a local UNIX or Linux user in the currently selected zone.
- save_local_group_profile saves the currently selected local UNIX or Linux group object after you create the group object or edit profile field values in the group object.
- save_local_user_profile saves the currently selected local UNIX or Linux user object after you create the user object or edit profile field values in the user object.
- select_local_group_profile selects a local UNIX or Linux group object for viewing or editing.
- select_local_user_profile selects a local UNIX or Linux user object for viewing or editing.
- set_local_group_profile_field sets the value of a field for the currently selected local UNIX or Linux group that has a profile defined in the current zone.

set_object_field

Use the set_object_field command to set the value for a specified field in the currently selected Active Directory object stored in memory. The set_object_field command does *not* set a field value stored in Active Directory for this object.

If you change any fields, you must save the object using the save_object command for your changes to take effect in Active Directory. If you select another object or end the ADEdit session before saving the currently selected object, your changes will be lost.

The set_object_field command does not check to see if fields and values are valid. When you save an object, Active Directory will check fields and values at that time and report an error if they aren't valid.

Zone Type

Not applicable

Syntax

```
set_object_field field value
```

Abbreviation

sof

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field you want to set. The *field* argument can by any attribute that is valid for the type of Active Directory object currently selected in memory.
value		Required. Specifies the value to assign to the specified field. The data type depends on the specified field. The set_object_field command does not check whether the value is valid. Active Directory will check for valid values when ADEdit saves the object.

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_object_field sd $sdvalue
```

This example sets the current object's security descriptor field to the string contained in the variable sdvalue (an SDDL string).

Related Commands

The following commands enable you to view and select Active Directory objects:

- get_objects performs an LDAP search of Active Directory and returns a Tcl list of the distinguished names of objects matching the specified search criteria.
- new_object creates a new Active Directory object and stores it in memory.
- select_object retrieves an object with its attributes from Active Directory and stores it in memory.

After you have an object stored in memory, you can use the following commands to work with that object:

- add_object_value adds a value to a multi-valued field attribute of the currently selected Active Directory object.
- delete_object deletes the selected Active Directory object from Active Directory and from memory.
- delete_sub_tree deletes an Active Directory object and all of its children from Active Directory.
- get_object_field reads a field value from the currently selected Active Directory object.
- remove_object_value removes a value from a multi-valued field attribute of the currently selected Active Directory object.
- save_object saves the selected Active Directory object with its current settings to Active Directory.

set_pam_field

Use the set_pam_field command to set the value for a specified field in the currently selected PAM application right stored in memory. The set_pam_field command does *not* set a field value stored in Active Directory for this PAM application right.

If you change any fields, you must save the PAM application right using the save_pam_app command for your changes to take effect in Active Directory. If you select another PAM application right or end the ADEdit session before saving the currently selected PAM application right, your changes will be lost.

You can only use the set_pam_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

```
set_pam_field field value
```

Abbreviation

spf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field that you want to set. The possible values are: **application**: The name of the PAM application that is allowed to use the adclient PAM authentication service. The name can be literal, or it can contain ? or * wildcard characters to specify multiple applications. **description**: Text describing the PAM application. Note that in a classic zone, setting the **application** field changes the name of the PAM application right. For example, assume you create a new PAM application right in a classic zone using a command like this: new_pam_app myftp If you then use this command to set the **application** field like this: set_pam_field application newftp The PAM application right itself will be renamed. If you were to use the list_pam_apps command after running the set_pam_field command, the right would be returned as newftp: list_pam_apps newftp : Renamed application right
value		Required. Specifies the value to assign to the specified field. In most cases, you can assign an empty string to unset a field value.

Return Value

This command returns nothing if it runs successfully.

Examples

set_pam_field application *

This example sets the application field for the current PAM application right to allow PAM access rights to all applications (* is the wildcard for all possible strings).

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select PAM application rights:

- get_pam_apps returns a Tcl list of PAM application rights in the current zone.
- list_pam_apps lists to stdout the PAM application rights in the currently selected zone.
- new_pam_app creates a new PAM application right and stores it in memory.
- select_pam_app retrieves a PAM application right from Active Directory and stores it in memory.

After you have a PAM application right stored in memory, you can use the following commands to work with that PAM application right:

- delete_pam_app deletes the selected PAM application right from Active Directory and from memory.
- get_pam_field reads a field value from the currently selected PAM application right.
- save_pam_app saves the selected PAM application right with its current settings to Active Directory.

set_role_assignment_field

Use the set_role_assignment_field command to sets the value for a specified field in the currently selected role assignment stored in memory. The set_role_assignment_field command does *not* set a field value stored in Active Directory for this role assignment.

If you change any fields, you must save the role assignment using the save_role_assignment command for your changes to take effect in Active Directory. If you select another role assignment or end the ADEdit session before saving the currently selected role assignment, your changes will be lost.

You can only use the set_role_assignment_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

set_role_assignment_field field value

Abbreviation

sraf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field that you want to set. The possible values are: **customAttr**: Sets custom text strings for the role assignment. This field is only applicable for hierarchical zones. **description**: Sets the description for the role assignment. **from**: Sets the starting date and time for the role assignment. The date and time is expressed in standard UNIX time. The Tcl clock command manipulates these time values. A value of 0 means no starting date and time for the role assignment. **role**: Sets the name of the role to assign and the zone in which the role was defined.The zone value is optional if the role is defined in the currently selected zone. The zone is required if the role is defined in another zone. **to**: Sets the ending date and time for the role assignment. The start and end dates and times are expressed in standard UNIX time. You can use the Tcl clock command to manipulate these values. A value of 0 indicates no date or time is set for the role assignment.
value	depends on field	Required. Specifies the value to assign to the specified field. In some cases, you can assign a dash (-) or an empty string to unset a field value. However, this is not supported for all fields or all zone types.

Return Value

This command returns nothing if it runs successfully.

Examples

set_role_assignment_field role su-root/global

This example assigns the role named su-root that is defined in the global zone.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a role assignment:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- list_role_assignments lists to stdout the role assignments in the current zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the currently selected role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- write_role_assignment saves the selected role assignment to a file.

set_role_field

Use the set_role_field command to set the value for a specified field in the currently selected role stored in memory. The set_role_field does *not* set a field value stored in Active Directory for this role.

If you change any fields, you must save the role using the save_role command for your changes to take effect in Active Directory. If you select another role or end the ADEdit session before saving the currently selected role, your changes will be lost.

You can only use the set_role_field command if the currently selected zone is a classic4 or hierarchical zone. The command does not work in other types of zones.

Zone Type

Classic and hierarchical

Syntax

set_role_field field value

Abbreviation

srf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field that you want to set.
value		Required. Specifies the value to assign to the specified field. In most cases, you can assign an empty string or null value (0) to unset a field value, depending on the data type of the field.

The data type required depends on the field you are setting. The possible values are:

- **allowLocalUser**: Set the value to true to allow local users to be assigned to the role, or false if local users should not be assigned to the role. This field is not applicable in classic zones. The valid values are 1, y, yes, or true to enable or 0, n, no, or false to disable. All other values throw an exception.
- **AlwaysPermitLogin**: Set the value to true to enable "rescue rights" for users assigned to the role, or false if "rescue rights" should not be applied to the role. This field is not applicable in classic zones. The valid values are 1, y, yes, or true to enable or 0, n, no, or false to disable. All other values throw an exception.
- **auditLevel**: Set the value to one of the following to specify whether auditing is not requested, requested but not required, or required:
  - AuditIfPossible
  - AuditNotRequested
  - AuditRequired

  This field is not applicable in classic zones.
- **customAttr**: Sets custom text strings for the role. This field is only applicable for hierarchical zones.
- **description**: Set the value to a text string that describes the role.
- **sysrights**: Set the value to specify the system rights granted to the role. This value is an integer that represents a combination of binary flags, one for each right. This field is not applicable in classic zones.
- **timebox**: Set the value to indicate the hours in the week when the role is enabled. This value is a 42-digit hexadecimal number. When represented in binary, each bit represents an hour of the week as described in the appendix Timebox Value Format

- **visible**: Returns true or false depending on whether "User is visible" right is configured for the role. You cannot get this field value if the selected zone is a classic zone.

Setting the system rights field value for a role

You can specify the sysrights field to define the system rights that you want to grant to the currently selected role. This field value is an integer that represents a combination of binary flags, with one flag for each of the following system rights:

1—Password login and non password (SSO) login are allowed.

2—Non password (SSO) login is allowed.

4—Account disabled in Active Directory can be used by sudo, cron, etc.

8—Log in with non-restricted shell.

16—Audit not requested/required.

32—Audit required.

64—Always permit to login.

128—Remote login access is allowed for Windows computers.

256—Console login access is allowed for Windows computers.

512—Require multi-factor authentication through the Delinea Connector to log on.

1024—PowerShell remote access is allowed

These values are added together to define the sysrights field value. For example, a sysrights value of 6 indicates that the role is configured to allow single sign-on login and to ignore disabled accounts (2+4). A value of 11 indicates that most common UNIX system rights are enabled (1+2+8). A value of 384 indicates that most common Windows system rights are enabled (128+256).

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the system rights for the current role to allow SSO login (2) and to provide a full shell (8):

```
set_role_field sysrights 10
```

The following example sets the current role to require auditing:

```
set_role_field auditLevel AuditRequired
```

Note that the sysrights field is a bit field, so you can add and remove bits for the field instead of setting the integer value directly. For example to add the system rights for single sign-on and full shell to existing system rights, you might use commands similar to this:

```
set sr [get_role_field sysrights]

set_role_field sysrights [expr { $sr | 10 }]
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select roles:

- get_roles returns a Tcl list of roles in the current zone.
- list_roles lists to stdout the roles in the current zone.
- new_role creates a new role and stores it in memory.
- select_role retrieves a role from Active Directory and stores it in memory.

After you have a role stored in memory, you can use the following commands to work with that role:

- add_command_to_roleadds a UNIX command to the current role.
- add_pamapp_to_role adds a PAM application right to the current role.
- delete_role deletes the selected role from Active Directory and from memory. *get_role_apps returns a Tcl list of the PAM applications associated with the currently selected role.
- get_role_commands returns a Tcl list of the UNIX commands associated with the current role.
- get_role_field reads a field value from the currently selected role.
- list_role_rights returns a list of all UNIX commands and PAM application rights associated with the current role. *remove_command_from_roleremoves a UNIX command from the current role.
- remove_pamapp_from_role removes a PAM application from the current role.
- save_role saves the selected role with its current settings to Active Directory.

set_rs_env_for_role

Use the set_rs_env_for_role command to assign a restricted shell environment to the currently selected role that is stored in memory. You should note that a role can only have one restricted shell environment assigned to it. If you assign a new restricted shell environment to a role, the current restricted shell environment—if one exists—will be removed. In addition, a role cannot be defined with both privileged commands and a restricted shell environment at the same time. If you assign a restricted shell environment to the currently selected role, all privileged commands previously defined for the role—if they exist—will be removed from the role.

The set_rs_env_for_role command does not modify the data stored in Active Directory for the restricted shell environment. If you run this command using ADEdit without saving the role to Active Directory, your changes do not take effect.

You can only use the set_rs_env_for_role command if the currently selected zone is a classic4 zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

set_rs_env_for_role environment

Abbreviation

srse

Options

This command takes no options.

Arguments

This command takes the following argument:

environment	string	Required. Specifies the name of the restricted shell environment to assign to the current role.

Return Value

This command returns nothing if it runs successfully.

Examples

set_rs_env_for_role rse1

This example sets the currently selected role's restricted shell environment to rse1, and removes any existing restricted shell environment or privileged commands if they exist in the role.

Related Commands

The following commands perform actions related to this command:

- clear_rs_env_from_role removes a restricted shell environment from the current role.
- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with that: restricted shell environment:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.

set_rsc_field

Use the set_rsc_field command to set the value for a specified field for the currently selected restricted shell command that is stored in memory. The set_rsc_field command does not set the field value stored in Active Directory for the selected restricted command field.

If you change any fields, you must save the restricted shell command using the save_rs_command command for your changes to take effect in Active Directory. If you select another restricted shell command or end the ADEdit session before saving the currently selected restricted shell command, your changes will be lost.

You can only use the set_rsc_field command if the currently selected zone is a classic4 zone is the selected zone. The command does not work in other types of zones.

Zone Type

Classic only

Syntax

set_rsc_field field value

Abbreviation

srscf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field whose value you want to set.
value		Required. Specifies the value you want to assign to the specified field. The data type depends on the field specified. In most cases, you can assign an empty string or null value (0) to unset a field value, depending on the data type of the field.

The possible field values are:

- **description**: Text describing the restricted shell command.
- **cmd**: The restricted shell command string or strings. You can use wild cards or a regular expression.
- **path**: The path to the command's location. You can use wild cards or a regular expression.
- **form**: An integer that indicates whether the cmd and path strings use wild cards (0) or a regular expression (1).
- **dzsh_runas**: A list of users and groups that can run this command in a restricted shell environment (dzsh). Users can be listed by user name or UID.

- **keep**: A comma-separated list of environment variables from the current user's environment to keep.
- **del**: A comma-separated list of environment variables from the current user's environment to delete.
- **add**: A comma-separated list of environment variables to add to the final set of environment variables.
- **pri**: An integer that specifies the command priority for the restricted shell command object.
- **umask**: An integer that defines who can execute the command.
- **flags**: An integer that specifies a combination of different properties for the command.
- **createTime**: The time and date this command was created, returned in generalized time format.
- **modifyTime**: The time and date this command was last modified, returned in generalized time format.
- **dn**: The command's distinguished name.

Setting the cmd and path field values for a restricted command

You can specify the cmd and path strings using wild cards (*, ?, and !), or as a regular expression. If you specify the cmd and path strings using wild cards, use an asterisk (*) to match zero or more characters, the question mark (?) to match exactly one character, or the exclamation mark (!) to negate matching of the specified string.

For both the cmd and path fields, the form field controls whether the specified string is interpreted as a regular expression or as a string that includes wild cards.

Specifying the environment variables for a restricted command

You can use the keep, del, and add settings to control the environment variables used by the commands specified by the cmd string. The keep and del settings are mutually exclusive. The keep field only takes effect if the flag 16 is included in the setting for the flag field. The del field only takes effect if the flag 16 is not included in the setting for the flag field.

Any environment variables kept or deleted are in addition to the default set of the user's environment variables that are either retained or deleted. The default set of environment variables to keep is defined in the dzdo.env_keep configuration parameter in the centrifydc.conf file. The default set of environment variables to delete is defined in the dzdo.env_delete configuration parameter in the centrifydc.conf file. You can also add environment variables to the final set of environment variables resulting from the keep or del fields.

Specifying the restricted command priority

You can use the pri field to specify the command priority when there are multiple matches for the restricted shell command object specified by wild cards. If there are multiple commands specified by this restricted shell command object, the restricted shell command with the higher command priority prevails.

Specifying the umask value for restricted commands

You can use the umask field to define who can execute the command. The umask field specifies a 3-digit octal value that defines read, write, or execute permission for owner, group, and other users. The left digit defines the owner execution rights, the middle digit defines the group execution rights, and the right digit defines other execution rights. Each digit is a combination of binary flags, one flag for each right as follows:

- 4 is read
- 2 is write
- 1 is execute

You add these values add together to define the rights available for each entity. For example, a umask value of 600 indicates read and write permission (4+2) for the owner, but no permissions for the group or other users. Similarly, a umask value of 740 indicates read, write, execute permissions (4+2+1) for the owner, read permissions for the group, but no permissions for other users.

Specifying restricted command properties using the flags field

You can use the **flags** field to define a combination of binary flags, with one flag for each of the following properties:

- 1 to prevent nested command execution. If this flag value is not set, nested command execution is allowed.
- 2 to require authentication with the user's password. You cannot set this flag and the 4 flag simultaneously. If neither 2 nor 4 is set, authentication is not required.
- 4 to require authentication with the run-as user's password

  If you do not set the 2 flag or the 4 flag, authentication is not required.

- 8 to preserve group membership. If this flag value is not set, group membership is not preserved.
- 16 to reset environment variables for the command, deleting the variables specified in the dzdo.env_delete parameter and keeping the variables specified in the keep field. If this flag is not set, the command removes the unsafe environment variables specified in the dzdo.env_delete parameter along with any additional environment variables specified by the del field

You add these values together to define the setting for the flags field. For example, a flags field value of 5 prevents nested command execution and requires authentication using the run-as user's password (1+4).

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_rsc_field description {This is the restricted command description}
```

This example sets the current restricted shell command description field to the "This is the restricted command description" text string.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select the restricted shell command to work with:

- get_rs_commands returns a Tcl list of restricted shell commands in the current zone.
- list_rs_commands lists to stdout the restricted shell commands in the current zone.
- new_rs_command creates a new restricted shell command and stores it in memory.
- select_rs_command retrieves a restricted shell command from Active Directory and stores it in memory.

After you have a restricted shell command stored in memory, you can use the following commands to work with that restricted shell command:

- delete_rs_command deletes the selected command from Active Directory and from memory.
- get_rsc_field reads a field value from the currently selected command.
- save_rs_command saves the selected command with its current settings to Active Directory.

set_rse_field

Use the set_rse_field command to set the value for a specified field in the currently selected restricted shell environment that is stored in memory. The set_rse_field command does not set the field value stored in Active Directory for this restricted shell environment.

This command only sets the field value that is stored in memory. You must save the restricted shell environment using the save_rs_env command for your changes to take effect in Active Directory. If you select another restricted shell environment or end the ADEdit session before saving the currently selected restricted shell environment, your changes will be lost.

You can only use the set_rse_field command if the currently selected zone is a classic4 zone. The command does not work in other type of zones.

Zone Type

Classic only

Syntax

```
set_rse_field field value
```

Abbreviation

srsef

Options

This command takes no options.

Arguments

This command takes the following argument:

field	string	Required. Specifies the name of the field whose value you want to set. The only possible value is: **description**: Text describing the restricted shell environment.
value	depends on field	Required. Specifies the value to assign to the specified field. In most cases, you can assign an empty string to unset a field value.

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_rse_field description {This string is the restricted shell description}
```

This example sets the description field for the current restricted shell environment to the "This string is the restricted shell description" text string.

Related Commands

Before you use this command, you must have a currently selected role stored in memory. The following commands enable you to view and select the role to work with restricted shell environments:

- get_rs_envs returns a Tcl list of restricted shell environments.
- list_rs_envs lists to stdout the restricted shell environments.
- new_rs_env creates a new restricted shell environment and stores it in memory.
- select_rs_env retrieves a restricted shell environment from Active Directory and stores it in memory.

After you have a restricted shell environment stored in memory, you can use the following commands to work with its fields:

- delete_rs_env deletes the current restricted shell environment from Active Directory and from memory.
- get_rse_field reads a field value from the current restricted shell environment.
- save_rs_env saves the restricted shell environment to Active Directory.

set_sd_owner

Use the set_sd_owner command to set the owner of a security descriptor (SD). This command requires you to specify the security descriptor in SDDL (security descriptor definition language) form and the security identifier (SID) of the owner. The command sets and returns the updated security descriptor in SDDL form with the new owner.

Zone Type

Not applicable

Syntax

```
set_sd_owner sddl_string owner_sid
```

Abbreviation

```
sso
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

sddl_string	string	Required. Specifies a security descriptor in SDDL format.
owner_sid	string	Required. Specifies the security identifier (SID) of the owner to set.

Return Value

This command returns an security descriptor in SDDL format if it runs successfully. The security descriptor contains the new owner set by the command.

Examples

This example sets a new owner for a security descriptor. The security descriptor is the first long string after the command. The SID of the new owner is the much shorter string at the end of the command (shown in **boldface**).

set_sd_owner O:DAG:DAD:AI (A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA) (OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO) (A;;RCLCRPLO;;;AU)(OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015e5f28;;AO) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RCLCRPLO;;bf967a9c-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS) (A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA) (A;CIID;LC;;;RU) (A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA) **S-1-5-21-1076040321-332654908-468068287-1109***

This example returns the updated security descriptor:

O:S-1-5-21-1076040321-332654908-468068287-1109G:DAD:AI (A;;RCWDWOCCDCLCSWRPWPLOCR;;;DA) (OA;;CCDC;bf967aba-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967a9c-0de6-11d0-a285-00aa003049e2;;AO) (OA;;CCDC;bf967aa8-0de6-11d0-a285-00aa003049e2;;PO) (A;;RCLCRPLO;;;AU) (OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015e5f28;;AO) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;4c164200-20c0-11d0-a768-00aa006e0529;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;59ba2f42-79a2-11d0-9020-00c04fc2d3cf;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RP;037088f8-0ae1-11d2-b422-00a0c968f939;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967a86-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;bf967aba-0de6-11d0-a285-00aa003049e2;ED) (OA;CIIOID;RCLCRPLO;;4828cc14-1437-45bc-9b07-ad6f015e5f28;RU) (OA;CIIOID;RCLCRPLO;;bf967a9c-0de6-11d0-a285-00aa003049e2;RU) (OA;CIIOID;RCLCRPLO;;bf967aba-0de6-11d0-a285-00aa003049e2;RU) (OA;CIID;RPWPCR;91e647de-d96f-4b70-9557-d63ff4f3ccd8;;PS) (A;CIID;SDRCWDWOCCDCLCSWRPWPDTLOCR;;;EA) (A;CIID;LC;;;RU) (A;CIID;SDRCWDWOCCLCSWRPWPLOCR;;;BA)

Related Commands

The following commands perform actions related to this command:

- explain_sd converts an SD in SDDL format to a human-readable form.
- remove_sd_ace removes an access control entry (ACE) from an SD.
- add_sd_ace adds an access control entry to an SD.

set_user_password

Use the set_user_password command to set a new password for an Active Directory user or computer in Active Directory.

Zone Type

Not applicable

Syntax

set_user_password UPN password

Abbreviation

sup

Options

This command takes no options.

Arguments

This command takes the following arguments:

UPN	string	Required. Specifies the user principal name (UPN) of the user or computer whose password will be reset.
password	string	Required. Specifies the text string to set as the new password. If the string contains characters that might be misinterpreted by ADEdit's Tcl interpreter ($, for example), enclose the string in braces so that all characters are interpreted literally with no substitutions.

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_user_password adam.avery@acme.com {B4uC$work}
```

This example sets the password for adam.avery@acme.com to B4uC$work.

Related Commands

None.

set_zone_computer_field

Use the set_zone_computer_field command to set the value for a specified field in the currently selected zone computer stored in memory. The set_zone_computer_field command does *not* set a field value stored in Active Directory for this zone computer.

If you change any fields, you must save the zone computer using the save_zone_computer command for your changes to take effect in Active Directory. If you select another zone computer or end the ADEdit session before saving the currently selected zone computer, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
set_zone_computer_field field value
```

Abbreviation

szcf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field whose value want set. The possible values are: **cpus**: Set to a positive integer for the number of CPUs in the computer. **enabled**: Set the value to 1, y, yes, or true if the computer is enabled in the zone or to 0, n, no, or false if the computer is not enabled in the zone. All other values throw an exception. **licensetype**: Specifies the type of license a computer uses. The valid values for this field are server or workstation.
value		Required. Specifies the value to assign to the specified field. In some cases, you can assign a dash (-) to a field to unset the field value. However, this is not supported for all fields or all zone types.

Return Value

This command returns nothing if it runs successfully.

Examples

```
set_zone_computer_field cpus 2
```

This example sets the current zone computer's number of CPUs to 2.

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and manage the zone computers:

- get_zone_computers returns a Tcl list of the Active Directory names of all zone computers in the current zone.
- list_zone_computers lists to stdout the zone computers in the current zone.
- new_zone_computer creates a new zone computer and stores it in memory.
- select_zone_computer retrieves a zone computer from Active Directory and stores it in memory.

After you have a zone computer stored in memory, you can use the following commands to work with that zone computer:

- delete_zone_computer deletes the zone computer from Active Directory and from memory.
- get_zone_computer_field reads a field value from the currently selected zone computer.
- save_zone_computer saves the zone computer with its current settings to Active Directory.
- set_zone_computer_field sets a field value in the currently selected zone computer.

set_zone_field

Use the set_zone_field command to set the value for a specified field in the currently selected zone stored in memory. The set_zone_field command does *not* set a field value stored in Active Directory for the selected zone.

If you change any fields, you must save the zone using the save_zone command for your changes to take effect in Active Directory. If you select another zone or end the ADEdit session before saving the currently selected zone, your changes will be lost.

This command is not applicable if the currently selected zone is a classic-computer zone. You cannot set zone field values for classic-computer zones.

Zone Type

Classic and hierarchical

Syntax

```
set_zone_field field value
```

Abbreviation

szf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field that you want to set.
value		Required. Specifies the value to assign to the specified field. In most cases, you can assign an empty string to unset the field value. For more information about the values set by the zone fields, see the Field value section.

The data type required depends on the field you are setting. The possible field values are:

- **availableshells**: Sets the list of shells available to choose from when adding new users to the zone.
- **block.parent.zgroup**: Sets the value of the block.parent.zgroup field in the zone object's description.
- **cloudurl**: Sets the URL of the cloud instance associated with the selected zone.
- **computers**: Sets the UPN of the computer group assigned to the selected computer role.
- **customAttr**: Sets custom text strings for the zone. This field is only applicable for hierarchical zones.
- **defaultgid**: Sets the default primary group to assign to new users.
- **defaultgecos**: Sets the default GECOS data to assign to new users.
- **defaulthome**: Sets the default home directory to assign to new users.
- **defaultshell**: Sets the default shell to assign to new users.
- **description**: Sets the text string that describes the zone.
- **gidnext**: Sets the next GID to use when auto-assigning GID numbers to new groups.
- **gidreserved**: Sets the GID number or range of numbers (1-100) that are reserved.
- **groupname**: Sets the default group name used for new groups in the zone.
- **nisdomain**: Sets the name of the NIS domain for NIS clients to use.
- **nssvar**: Sets the NSS substitution variable to add to the zone's list of substitution variables.
- **parent**: Sets the distinguished name of the zone's parent zone.
- **sfudomain**: Sets the Windows domain name for the SFU zone.
- **sid2iddomainmap**: Sets the domain ID map for the seleted zone. Specify the mapping with a comma-separated key value pairs string. See the examples section for a sample command with this field. Note that the range of domain IDs is 0 to 511. Duplicate mapping entries are not allowed (domain names are not case-sensitive). This field is not supported for auto zones nor classic zones.
- **tenantid**: Returns the Delinea Platform tenant ID for the zone. This field is only applicable for hierarchical zones.
- **uidnext**: Sets the next UID to use when auto-assigning UID numbers to new users.
- **uidreserved**: Sets the UID number or range of numbers (1-100, for example) that are reserved.
- **username**: Sets the default user name used for new users in the zone.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the computer group associated with the currently selected computer role to linux_machines in the domain acme.com:

```
set_zone_field computers linux_machines@acme.com
```

The following example sets the parent zone of the current zone to global in the domain acme.com:

```
szf parent “CN=global,CN=zones,CN=Acme,CN=Program Data,DC=acme,DC=com”
```

The following example sets the domain ID mapping for the selected zone:

```
set_zone_field sid2iddomainmap domain0.test=0,domain1.test=1,domain2.test=2
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone to work with:

- create_zone creates a new zone in Active Directory.
- get_zones returns a Tcl list of all zones within a specified domain.
- select_zone retrieves a zone from Active Directory and stores it in memory.

After you have a zone stored in memory, you can use the following commands to work with that zone:

- delegate_zone_right delegates a zone use right to a specified user or computer.
- delete_zone deletes the selected zone from Active Directory and memory.
- get_child_zones returns a Tcl list of child zones, computer roles, or computer zones.
- get_zone_field reads a field value from the currently selected zone.
- get_zone_nss_vars returns the NSS substitution variable for the selected zone.
- save_zone saves the selected zone with its current settings to Active Directory.

set_zone_group_field

Use the set_zone_group_field command to set the value for a specified field in the currently selected zone group stored in memory. The set_zone_group_field command does *not* set a field value stored in Active Directory for the selected zone group.

If you change any fields, you must save the zone group using the save_zone_group command for your changes to take effect in Active Directory. If you select another zone group or end the ADEdit session before saving the currently selected zone group, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

set_zone_group_field field value

Abbreviation

szgf

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field that you want to set. The possible values are: **gid**: Sets the numeric identifier for the group (GID). **name**: Sets the text string for the group name. **required**: Specifies whether the zone group is required. Set the value to 1, y, yes, or true if the group is required in the zone or to 0, n, no, or false if the group in not required in the zone. All other values throw an exception.If a group is required, users cannot remove the group from their active set of groups. You can also specify AIX extended attributes as the field to set an extended attribute value for a group. Extended attribute fields start with the aix. prefix. For example, the admin extended attribute can be set by specifying aix.admin as the field.
value		Required. Specifies the value to assign to the specified field. The data type depends on the field specified. In some cases, you can assign a dash (-) to a field to unset the field value. However, this is not supported for all fields or all zone types.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the current zone group's UNIX group name to managers.

```
set_zone_group_field name managers
```

If the current group is on AIX, you can set AIX group extended attributes and values. For example, to identify the current group as an administrative group, you can set the admin extended attribute:

```
set_zone_group_field aix.admin true
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select zone groups:

- get_zone_groups returns a Tcl list of the Active Directory names of all zone groups in the current zone.
- list_zone_groups lists to stdout the zone groups in the current zone.
- new_zone_group creates a new zone group and stores it in memory.
- select_zone_group retrieves a zone group from Active Directory and stores it in memory.

After you have a zone group stored in memory, you can use the following commands to work with that zone group:

- delete_zone_group deletes the selected zone group from Active Directory and from memory.
- get_zone_group_field reads a field value from the current zone group.
- save_zone_group saves the selected zone group with its current settings to Active Directory.

set_zone_user_field

Use the set_zone_user_field command to set the value for a specified field in the currently selected zone user stored in memory. The set_zone_user_field command does *not* set a field value stored in Active Directory for this zone user.

If you use ADEdit to change any field, you must save the zone user using the save_zone_user command for your changes to take effect in Active Directory. If you select another zone user or end the ADEdit session before saving the currently selected zone user, your changes will be lost.

Zone Type

Classic and hierarchical

Syntax

```
set_zone_user_field field value
```

Abbreviation

```
szuf
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

field	string	Required. Specifies the name of the field y want set. The possible values are: **uname**: Sets the text string to use for the UNIX user name. If you are setting this field in a Service for UNIX (SFU) zone, this name must be unique among all the SFU zones. If you duplicate a user name that exists in another SFU zone, that user will be moved to the currently selected SFU zone when you save the zone user. **uid**: Sets the numeric identifier for the user (UID). **gid**: Sets the numeric identifier for the user's primary group (GID).Set the value to 0x80000000 to indicate a private group (the user's UID is used as the GID). **gecos**: Sets the text string to use for the user's GECOS field. **home**: Sets the text string that specifies the user's home directory. **shell**: Sets the text string that specifies the user's default shell type. **enabled**: Specifies whether user is enabled or not. This field is only valid in classic zones. Set the value to 1, y, yes, or true if the user is enabled in the zone or to 0, n, no, or false if the user is disabled in the zone. All other values throw an exception. You can also specify AIX extended attributes as the field to set an extended attribute value for a zone user.

value		Required. Specifies the value to assign to the specified field. The data type depends on the field specified. In some cases, you can assign a dash (-) to a field to unset the field value. However, this is not supported for all fields or all zone types.

Return Value

This command returns nothing if it runs successfully.

Examples

The following example sets the current zone user's UNIX user name to buzz:

```
set_zone_user_field uname buzz
```

This following example sets the current zone user's primary GID to the same value as the user's UID:

```
set_zone_user_field gid 0x80000000
```

If the current zone user is on AIX, you can set extended attributes and values. For example:

```
select_zone_user aixu1@acme.com set_zone_user_field aix.ttys u1,u2,u3 set_zone_user_field aix.fsize 209715 set_zone_user_field aix.core 2097151 set_zone_user_field aix.cpu -1
save_zone_user
```

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select a zone user:

- get_zone_users returns a Tcl list of the Active Directory names of all zone users in the current zone.
- list_zone_users lists to stdout the zone users and their NSS data in the current zone.
- new_zone_user creates a new zone user and stores it in memory.
- select_zone_user retrieves a zone user from Active Directory and stores it in memory.

After you have a zone user stored in memory, you can use the following commands to work with that zone user:

- delete_zone_user deletes the selected zone user from Active Directory and from memory.
- get_zone_user_field reads a field value from the currently selected zone user.
- save_zone_user saves the selected zone user with its current settings to Active Directory.

show

Use the show command to display the current context of ADEdit. The command shows the domains ADEdit is bound to, the objects that are currently selected, and all available data for each selected object as it is stored in memory.

You should note that the command returns stored object data as it currently exists in memory. If you use ADEdit commands to change objects, but have not yet saved the data back to Active Directory, the information returned by the show command will not match the object data stored in Active Directory.

Zone Type

Not applicable

Syntax

```
show [all|bind|zone|user|computer|assignment|object|group| pamright|dzcommand|nismap|role|license|rse|rscommand localuser|localgroup]
```

Abbreviation

None.

Options

This command takes no options.

Arguments

This command takes the following argument of type string:

[all | user | bind | zone | user | computer | assignment | object | group | pamright | dzcommand | nismap | role | license | rse | rscommand | localuser | localgroup]

You can limit the information returned by specifying one of the following arguments. If no argument is supplied, the default is **all**.

- **all** returns the complete context of ADEdit—all of its current bindings and all currently selected objects in memory.
- **bind** returns ADEdit's currently bound domains and the server bound in each domain.
- **zone** returns the currently selected zone.
- **user** returns the currently selected user object.
- **computer** returns the currently selected zone computer.
- **assignment** returns the currently selected role assignment
- **object** returns the currently selected Active Directory object.
- **group** returns the currently selected zone group.
- **pamright** returns the currently selected PAM application right.
- **dzcommand** returns the currently selected UNIX command.
- **nismap** returns the currently selected NIS map.
- **role** returns the currently selected role.
- **license** returns the forest list where valid licenses have been found (it only reports the forests that have been queried).
- **rse** returns the currently selected restricted shell environment.
- **rscommand** returns the currently selected restricted shell command.
- **localuser** returns the currently selected local user.
- **localgroup** returns the currently selected local group.

Return Value

This command returns domain bindings and/or object data, depending on the supplied argument.

Examples

show

This example returns information all bound domains and selected objects similar to this:

Bindings: acme.com: calla.acme.com Current zone: CN=global,CN=Zones,CN=Acme,CN=Program Data,DC=acme,DC=com Current nss user: adam.avery@acme.com:adam:10001:10001:%{u:samaccountname}:%{home}/%{user}:%{shell}:

Related Commands

None.

sid_to_escaped_string

Use the sid_to_escaped_string command to specify a security identifier (SID) and have it converted to an escaped string format that works in an LDAP filter.

Zone Type

Not applicable

Syntax

sid_to_escaped_string sid

Abbreviation

stes

Options

This command takes no options.

Arguments

This command takes the following argument:

sid	string	Required. Specifies a security identifier (SID).

Return Value

This command returns an escaped string form of the supplied security identifier.

Examples

```
sid_to_escaped_string S-1-5-21-2076040321-3326545908-468068287-1157
```

This example returns an escaped string:

```
\01\05\00\00\00\00\00\05\15\00\00\00\81\dc\bd\7b\f4\0f\47\c6\bf\27\e6\1b\85\04\00\00
```

Related Commands

The following commands perform actions related to this command:

- sid_to_uid converts an Active Directory security identifier to a user ID (UID).
- principal_from_sid searches Active Directory for an security identifier and returns the security principal associated with the security identifier.

sid_to_uid

Use the sid_to_uid command to specify a security identifier (SID) of an Active Directory user to look up the Active Directory user in Active Directory. This command converts the user's security identifier to a numeric identifier for the user ID (the UID value). This conversion process is the same process used to generate UIDs for Delinea Express users or when you use Auto Zone to automatically generate UIDs for users.

Zone Type

Not applicable

Syntax

```
sid_to_uid [-domainidmap] sid
```

Abbreviation

```
stu
```

Options

This command takes the following options:

-domainidmap	Optional. Specifies a domain ID mapping for the selected zone. Before using this field, you must have a selected zone stored in memory. This field is not supported for auto zones nor classic zones. If the selected zone does not already have a domain ID mapping, the UID is generated normally. If the selected zone has a domain ID mapping already and the domain to which this SID belongs exists in the specified domain ID mapping, the UID is generated with the algorithm based on the domain ID mapping. If the selected zone has a domain ID mapping already but the domain to which this SID belongs does not exist in the specified domain ID mapping, the UID is generated normally. For example: sid_to_uid -domainidmap S-1-5-21-2076040321-3326545908-468068287-1157

Arguments

This command takes the following argument:

sid	string	Required. Specifies a security identifier (SID).

Return Value

This command returns a numeric user ID.

Examples

This example returns a unique UID for the user: **1874853888**

Related Commands

The following commands perform actions related to this command:

- principal_from_sid searches Active Directory for an SID and returns the security principal associated with the SID.

validate_license

Use the validate_license command to specify a path to the Delinea license container and determine if there is a valid license. If there is a valid license, the command stores an indicator in the ADEdit current context. If the command does not find a valid license, it reports an error and exits.

ADEdit requires a valid license before a zone is created. The create_zone and create_computer_role commands do an implicit search for a valid license. For example, you can call create_zone and let it attempt to find the container and validate the license. If that command fails to find a valid license, use validate_license to validate the license container from an explicit path.

You can call the validate_license command multiple times. Successive indicators take precedence. The command writes separate indicators for each forest—that is, each license is valid for a forest. You can use the show license command to see the list of forests that have been found to have a valid license.

Do not call validate_license before you bind to the domain.

The validate_license context is deleted when ADEdit exits.

Zone Type

Not applicable

Syntax

validate_license path

Abbreviation

vl

Options

This command takes no options.

Arguments

This command takes the following argument:

path	string	Required. Specifies the path is the license container's distinguished name (DN).

Return Value

This command returns nothing.

Examples

validate_license “CN=Licenses,OU=Acme,DC=acme,DC=com”

This example looks in the acme.com/Acme/Licenses container for a valid license.

Related Commands

The following commands perform actions related to this command:

- bind defines the current domain.
- create_zone does in implicit validate license during execution.
- show with the license option lists all forests that have a valid license.

write_role_assignment

Use the write_role_assignment command to write the selected role assignment with its current settings to a file in JSON format. You can use this command to save the currently selected role assignment that is stored in memory to a file with the permissions of 0600. If the file already exists, the command truncates the file. If the command cannot open or write to the specified file, the command fails with an error message.

Zone Type

Hierarchical only

Syntax

write_role_assignment file

Abbreviation

wra

Options

This command takes no options.

Arguments

This command takes the following argument:

file	string	Required. Specifies the file to which the command writes the role assignment.

Return Value

This command returns nothing.

Examples

write_role_assignment roleassignment.txt

Related Commands

Before you use this command, you must have a currently selected zone stored in memory. The following commands enable you to view and select role assignment to work with:

- get_role_assignments returns a Tcl list of role assignments in the current zone.
- list_role_assignments lists to stdout the role assignments in the current zone.
- new_role_assignment creates a new role assignment and stores it in memory.
- select_role_assignment retrieves a role assignment from Active Directory and stores it in memory.
- save_role_assignment saves a role assignment to Active Directory.

After you have a role assignment stored in memory, you can use the following commands to work with that role assignment's attributes, delete the role assignment, or save information for the role assignment:

- delete_role_assignment deletes the selected role assignment from Active Directory and from memory.
- get_role_assignment_field reads a field value from the current role assignment.
- save_role_assignment saves the selected role assignment with its current settings to Active Directory.
- set_role_assignment_field sets a field value in the current role assignment.
- write_role_assignment saves the selected role assignment to a file.

ADEdit Tcl Procedure Library Reference

This chapter describes the commands in the ade_lib Tcl library. The command descriptions are in alphabetical order. The syntax of each command shows optional elements in [square brackets] and variables in *italics*.

add_user_to_group

Use the add_user_to_group command to add an Active Directory user to an Active Directory group.

Syntax

add_user_to_group *user group*

Options

This command takes no options.

Arguments

This command takes the following arguments:

user	string	Required. Specifies the user principal name (UPN) of the Active Directory user to add.
group	string	Required. Specifies the UPN of the Active Directory group to which to add the user.

Return value

This command returns nothing if it runs successfully.

Examples

```
add_user_to_group adam.avery@acme.com pubs@acme.com
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_aduser creates a new Active Directory user account and sets its password.
- create_adgroup creates a new Active Directory group account and specifies its scope.
- create_user creates a new zone user based on an existing Active Directory user, assigns field values to the new user, and saves the new user to Active Directory.
- create_group creates a new zone group based on an existing Active Directory group, assigns it a UNIX name and group ID, and saves the new group to Active Directory.
- remove_user_from_group removes an Active Directory user from an Active Directory group.

convert_msdate

Use the convert_msdate command to specify a Microsoft date value from an Active Directory object field such as pwdLastSet and convert it into a human-readable form.

Syntax

convert_msdate *msdate*

Options

This command takes no options.

Arguments

This command takes the following argument:

msdate	string	Required. Specifies the Microsoft date value for conversion.

Return value

This command returns the day of the week, the day of the month, the time of day using a 24-hour clock, the time zone, and the year.

Examples

```
convert_msdate [get_object_field pwdLastSet]
```

This example returns converted into a format similar to this:

Thu Mar 24 14:40:26 PDT 2010

The unseen value returned by `get_object_field pwdLastSet` in this example was 129140268240625000, which was converted to a human-readable time and date.

Related Tcl library commands

None.

create_adgroup

Use the `create_adgroup` command to create a new Active Directory group account with a specified distinguished name (DN), sAMAccountName, and group scope.

Syntax

```
create_adgroup dn sam gscope`
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

dn	string	Required. Specifies the distinguished name of the new group.
sam	string	Required. Specifies the sAMAccountName of the new group.
gscope	string	Required. Specifies the scope for the new group. The possible values are: global universal local (for Domain local)

Return value

This command returns nothing if it runs successfully.

Examples

```
create_adgroup {CN=pubs,CN=Users,DC=acme,DC=com} pubs global
```

This example creates the group pubs with a global scope in the Active Directory Users container.

```
create_adgroup {CN=ApacheAdmins,OU=Unix Groups,OU=ACME,DC=acme,DC=com} pubs global
```

This example creates the group ApacheAdmins in the organizational unit Unix Groups, which is in the organizational unit ACME.

Related Tcl library commands

The following commands perform actions related to this command:

- `create_aduser` creates a new Active Directory user account and sets its password.
- `create_user` creates a new zone user based on an existing Active Directory user, assigns field values to the new user, and saves the new user to Active Directory.
- `create_group` creates a new zone group based on an existing Active Directory group, assigns it a UNIX name and group ID, and saves the new group to Active Directory.
- `add_user_to_group` adds an Active Directory user to an Active Directory group.
- `remove_user_from_group` removes an Active Directory user from an Active Directory group.

create_aduser

Use the `create_aduser` command to create a new Active Directory user account with a specified distinguished name (DN), user principal name (UPN), sAMAccountName, and password.

Syntax

```
create_aduser dn upn sam pw ?dname? ?gname? ?spn? ?gecos?
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

dn	string	Required. Specifies the distinguished name of the new user.
upn	string	Required. Specifies the user principal name of the new user.
sam	string	Required. Specifies the sAMAccountName of the new user.
pw	string	Required. Specifies the password for the new user.
dname	string	Optional. Specifies the displayName for the new user.
gname	string	Optional. Specifies the givenName for the new user.
spn	string	Optional. Specifies the servicePrincipalName for the new user.
gecos	string	Optional. Specifies the gecos for the new user.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_aduser {CN=ulysses urkham,CN=Users,DC=acme,DC=com} ulysses.urkham@acme.com ulysses.urkham {5$6fEr2B}
```

This example creates a new Active Directory user account ulysses.urkham@acme.com.

Related Tcl library commands

- create_adgroup creates a new Active Directory group account and specifies its scope.
- create_user creates a new zone user based on an existing Active Directory user, assigns field values to the new user, and saves the new user to Active Directory.
- create_group creates a new zone group based on an existing Active Directory group, assigns it a UNIX name and group ID, and saves the new group to Active Directory.
- add_user_to_group adds an Active Directory user to an Active Directory group.
- remove_user_from_group removes an Active Directory user from an Active Directory group.

create_assignment

Use the create_assignment command to create a new role assignment for a user or group and save it to Active Directory.

Syntax

```
create_assignment upn role[/zonename] [from] [to] [description]
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

upn	string	Required. Specifies the user principal name of the Active Directory user or group to whom to assign the role.
role*/[zonename]*	string	Required. Specifies the name of the role to assign and (optional) the name of the zone in which the role is assigned. If the zone name is present, a slash(/) separates the role name and the zone name. If the zone name isn't present, the role assignment occurs in the currently selected zone.
from	string	Optional. Specifies the start date and time for the role assignment. The start date and time can be expressed using the format: yr-mon-day hour:min
to	string	Optional. Specifies the expiration date and time for the role is assignment. The expiration date and time can be expressed using the format: yr-mon-day hour:min
description	string	Optional. Specifies a description of the role assignment.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_assignment ulysses.urkham@acme.com admin/support {} {} "Test assignment"
```

This example creates a role assignment for the rights defined in the role "admin" from the "support" zone to the user Ulysses Urkham. The role assignment is set to start immediately (by specifying) and never expire (by specifying the second) and has an optional description.

```
create_assignment amy@example.demo mgr {2021-03-31 10:51} {2021-03-31 12:51}
```

This example creates a role assignment for the rights defined in the role "mgr" from the current zone to the user amy@example.com. This role assignment is set to start at a specific time and expire two hours later and has no description.

Related Tcl library commands

None.

create_dz_command

Use the create_dz_command command to create a new UNIX privileged command in the currently selected zone.

Syntax

```
create_dz_command dzc cmd ?desc? ?form? ?dzdo_runas? ?dzsh_runas? ?flags? ?pri? ?umask? ?path? ?selinux_role? ?selinux_type?
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

name	string	Required. Specifies the name to assign to the new UNIX command.
command	string	Required. Specifies the UNIX command string or strings. You can use wild cards or a regular expression.
description	string	Optional. Specifies text describing the UNIX command.
form	integer	Optional. Specifies whether the command and path strings use wild cards (0) or a regular expression (1).
dzdo_runas	string	Optional. Specifies the list of users and groups that can run this command under dzdo (similar to sudo). Users can be listed by user name or UID.
selinux_role	string	Optional. Specifies the SELinux role to use when constructing a new security context for command execution. Note that selinux_role is only supported on Red Hat Enterprise Linux systems and effective only on systems with SELinux enabled and joined to a hierarchical zone.
selinux_type	string	Optional. Specifies the SELinux type to use when constructing a new security context for command execution. Note that selinux_type is only supported on Red Hat Enterprise Linux systems and effective only on systems with SELinux enabled and joined to a hierarchical zone.
dzsh_runas	string	Optional. Specifies the list of users and groups that can run this command in the restricted shell environment (dzsh). Users can be listed by user name or UID.
flags	integer	Optional. Specifies an integer that defines a combination of different properties for the command. For more information about setting this field, see set_dzc_field.
pri	integer	Optional. Specifies the command priority for the restricted shell command object. For more information about setting this field, see set_dzc_field.
umask	integer	Optional. Specifies an integer that defines who can execute the command. For more information about setting this field, see set_dzc_field.
path	string	Optional. Specifies the path to the command's location. You can use wild cards, a regular expression, or one of the following keywords: USERPATH to set to the command path to the equivalent of the Standard user path option. SYSTEMPATH to set to the path to the equivalent of the Standard system path option. SYSTEMSEARCHPATH to set to the path to the equivalent of the System search path option. If you don't specify this argument, the default is USERPATH.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_dz_command testvi vi {Test UNIX command vi} {} {sfapps:perez,cody} {} {16}
```

Related Tcl library commands

None.

create_group

Use the create_group command to create a new zone group for the currently selected zone. This command creates the new group based on an existing Active Directory group. It also assigns the new group a new UNIX profile that includes the UNIX group name and the UNIX group numeric identifier (GID).

Syntax

```
create_group upn name gid ?req?
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

upn	string	Required. Specifies the user principal name of the Active Directory group to use as the basis for the new zone group.
name	string	Required. Specifies the UNIX group name of the new zone group. For hierarchical zones only, specifying "-" unsets the name value.
gid	string	Required. Specifies the UNIX group ID to assign to the new zone group. For hierarchical zones only, specifying "-" unsets the gid value.
req	string	Optional. Specifies whether the zone group is required. Set the value to 1, y, yes, or true if the group is required in the zone or to 0, n, no, or false if the group in not required in the zone. All other values throw an exception. If a group is required, users cannot remove the group from their active set of groups.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_group pubs@acme.com pubs 1094
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_aduser creates a new Active Directory user account and sets its password.
- create_adgroup creates a new Active Directory group account and specifies its scope.
- create_user creates a new zone user based on an existing Active Directory user, assigns field values to the new user, and saves the new user to Active Directory.
- add_user_to_group adds an Active Directory user to an Active Directory group.
- remove_user_from_group removes an Active Directory user from an Active Directory group.

create_nismap

Use the create_nismap command to create a new NIS map in the currently selected zone.

Syntax

```
create_nismap map key:value comment
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

map	string	Required. Specifies the name of the new NIS map
key	string	Required. Specifies the key of the NIS map entry.
value	string	Required. Specifies the value of the NIS map entry.
comment	string	Required. Specifies the comment for the NIS map entry.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_nismap animals {{cat:1 {The cat says "Mew\!".}} {cow:1 {The cow says "Moo\!".}}}
```

Related Tcl library commands

None.

create_pam_app

Use the create_pam_app command to create a new PAM application access right in the currently selected zone.

Syntax

```
create_pam_app name application description
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

name	string	Required. Specifies the name to assign to the new PAM application access right.
		Required. Specifies the name of the PAM application that is allowed to use the adclient PAM authentication service. The name can be literal, or it can contain ? or * wild card characters to specify multiple applications. Note that in a classic zone, setting the **application**

application	string	field changes the name of the PAM application right. For example, assume you create a new PAM application right in a classic zone using a command like this: create_pam_app myftp newftp "Sample PAM FTP application". The PAM application right itself will be renamed as newftp: list_pam_apps newftp: Sample PAM FTP application. Therefore, in a classic zone, you should always specify the same string for the name and application arguments. In a hierarchical zone, you can specify different strings for the arguments.
description	string	Optional. Specifies the text describing the PAM application.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_pam_app testvi vi {Test UNIX command vi}
```

Related Tcl library commands

None.

create_role

Use the create_role command to create a new role definition in the currently selected zone.

Syntax

```
create_role name description sysrights pamrights cmdrights allowlocal rsenv visible
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

name	string	Required. Specifies the name to assign to the new role.
description	string	Specifies the text string that describes the role.
sysrights	integer	Specifies the system rights granted to the role. This value is an integer that represents a combination of binary flags, one for each system right. This field is not applicable in classic zones.
pamrights[/zonename]	string	Specifies the PAM application rights to add to the currently selected role. If the PAM application right that you want to add is defined in the current zone, the *zonename* argument is optional. If the PAM application right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific PAM application right to add.
cmdrights[/zonename]	string	Specifies the UNIX command rights to add to the currently selected role. If the UNIX command right that you want to add is defined in the current zone, the *zonename* argument is optional. If the UNIX command right is defined in a zone other than the currently selected zone, the *zonename* argument is required to identify the specific UNIX command right to add.
allowlocal	Boolean	Specifies whether local users can be assigned to the role. If this argument is specified, local users can be assigned to the role. This argument is only applicable if the zone is a hierarchical zone.

rsenv	string	Specifies a restricted shell environment for the role you are creating. This argument is only applicable if the zone is a classic zone.
visible	Boolean	Specifies whether the account profiles for Active Directory users in the role are visible on computers in the zone. This argument is only applicable if the zone is a hierarchical zone.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_role dba {Database admins - US} 11 {{oracle} {ftp}} {{testvi} {ora-stp}}
```

Related Tcl library commands

None.

create_rs_command

Use the create_rs_command command to create a new restricted shell command for the currently selected restricted shell environment.

Syntax

```
create_rs_command rsc_name cmd description form dzsh_runas flags pri umask path
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

rsc_name	string	Required. Specifies the name of the restricted shell command.
cmd	string	Required. Specifies the restricted shell command string or strings. You can use wild cards or a regular expression.
description	string	Optional. Specifies the text describing the restricted shell command.
form	integer	Optional. Indicates whether the cmd and path strings use wild cards (0) or a regular expression (1).
dzsh_runas	string	Optional. Specifies the list of users and groups that can run this command in a restricted shell environment (dzsh). Users can be listed by user name or UID.
flags	string	Optional. Specifies an integer that specifies a combination of different properties for the command. For more information about setting this field, see set_rsc_field.
pri	integer	Optional. Specifies the command priority for the restricted shell command object. For more information about setting this field, see set_rsc_field.
umask	integer	Optional. Specifies an integer that defines who can execute the command. For more information about setting this field, see set_rsc_field.
		Optional. Specifies the path to the restricted command. You can use wild cards, a regular expression, or one of the following

path	string	keywords: USERPATH to set to the command path to the equivalent of the Standard user path option. SYSTEMPATH to set to the path to the equivalent of the Standard system path option. SYSTEMSEARCHPATH to set to the path to the equivalent of the System search path option. If you don't specify this argument, the default is USERPATH.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_rs_command test_id id {Sample restricted command description}
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_rs_env creates a new restricted shell environment.

create_rs_env

Use the create_rs_env command to create a new restricted shell environment for the currently selected zone.

Syntax

```
create_rs_env rse_name rse_description
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

rse_name	string	Required. Specifies the name of the new restricted shell environment.
rse_description	string	Optional. Specifies the description for the new restricted shell environment.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_rs_env restrictedenv “This is a restricted shell environment”
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_rs_command creates a new restricted shell command.

create_user

Use the create_user command to create a new zone user for the currently selected zone. This command creates the new user based on an existing Active Directory user. It also assigns the new user a new UNIX profile that includes the user name, user ID, primary group ID, GECOS data, home directory, shell type,

and role (or in classic zones whether the user is enabled or disabled).

You can assign the new user a role in a non-classic zone or you can enable or disable the new user in a classic zone. In a non-classic zone, create_user uses whatever role you specify to create a new role assignment object that links the new zone user to the specified role.

Syntax

```
create_user UPN uname uid gid gecos home shell role
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

UPN	string	Required. Specifies the user principal name of the Active Directory user to use as the basis for the new zone user.
uname	string	Required. Specifies the user name of the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the user name.
uid	string	Required. Specifies the user ID for the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the user ID.
gid	string	Required. Specifies the group ID for the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the group ID.
gecos	string	Required. Specifies the GECOS value (new user account information) for the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the GECOS value. You can't set the GECOS value if the currently selected zone is a classic zone.
home	string	Required. Specifies the home directory for the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the home directory.
shell	string	Required. Specifies the shell type for the new zone user. For hierarchical zones, you can specify a dash (-) for this argument if you don't want to set the shell type.
role	string or Boolean value	Required. For classic zones, this argument determines whether to enable or disable the new zone user. A value of 1, Y, or y enables the user. Any other value disables the user. For hierarchical zones, this argument identifies the role to assign to the new zone user. You can specify a dash (-) for this argument if you don't want to set the role. However, a role must be assigned before the new zone user has access to computers in hierarchical zones.

Return value

This command returns nothing if it runs successfully.

Examples

```
create_user ulysses.urkham@acme.com ulysses 1005 - - %{home}/%{user} %{shell} -
```

This example creates a zone user "ulysses" based on the Active Directory user ulysses.urkham@acme.com. It sets a UID, does not set a GID or GECOS value by using dashes, sets home and shell values, and does not set a role value (specified by using a dash).

Related Tcl library commands

- create_aduser creates a new Active Directory user account and sets its password.

- create_adgroup creates a new Active Directory group account and specifies its scope.
- create_group creates a new zone group based on an existing Active Directory group, assigns it a UNIX name and group ID, and saves the new group to Active Directory.
- add_user_to_group adds an Active Directory user to an Active Directory group.
- remove_user_from_group removes an Active Directory user from an Active Directory group.

decode_timebox

Use the decode_timebox command to convert an internal timebox value that defines when a role is enabled or disabled into a format that can be evaluated. The command converts the internal hexadecimal value for a role timebox to a hexadecimal timebox value format as described in **Timebox Value Format**.

The command returns a 168-bit value in hexadecimal format that delineates the hours of the week from midnight Sunday to 11 PM Saturday in order from most-significant bit to least-significant bit. If a bit is set to 1, its corresponding hour is enabled for the role. If set to 0, its corresponding hour is disabled.

This command is useful for deciphering the value returned by the get_role_field for the timebox field.

Syntax

```
decode_timebox strTimeBox
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

strTimeBox	hex	A 42-digit hexadecimal timebox value. A value of zero disables all hours of the week. A value of FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF enables all hours of the week.

Return value

This command returns a decoded hexadecimal value that is the timebox value for a role.

Examples

```
>select_role test1
>get_role_field timebox
FFF7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
>package require ade_lib
1.0
>decode_timebox [grf timebox]
```

This example returns the decoded 42 hexadecimal that indicates the role is disabled from midnight to one on Sunday:

7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Related Tcl library commands

The following commands perform actions related to this command:

- encode_timebox converts a readable timebox value to an internal timebox format.
- modify_timebox defines an hour of the week and enables or disables that hour in the timebox value.

encode_timebox

Use the encode_timebox command to convert a human-readable timebox value that defines the when a role is enabled or disabled to an internal timebox value format.

The command converts the hexadecimal timebox value format described in **Timebox Value Format** to the internal hexadecimal value for a role. The command accepts a 168-bit value in hexadecimal format that delineates the hours of the week from midnight Sunday to 11 PM Saturday from most-significant bit to leastsignificant bit. If a bit is set to 1, its corresponding hour is enabled for the role. If set to 0, its corresponding hour is disabled.

This command is useful for setting the timebox field with the set_role_field command.

Syntax

```
encode_timebox strTimeBox
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

strTimeBox	hex	A 42-digit hexadecimal timebox value. A value of zero disables all hours of the week. A value of FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF enables all hours of the week.

Return value

This command returns a decoded hexadecimal value that is the timebox value for a role.

Examples

```
>package require ade_lib
>set tb 7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
>encode_timebox \$tb
```

This example returns the encoded 42 hexadecimal that indicates the role is disabled from midnight to one on Sunday:

```
FFF7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Related ade_lib Tcl library commands

The following commands perform actions related to this command:

- decode_timebox converts an internal timebox value to a decipherable format.
- modify_timebox defines an hour of the week and enables or disables that hour in the timebox value.

explain_groupType

Use the explain_groupType command to convert a groupType value from an Active Directory object field into human-readable form.

Syntax

```
explain_groupType gt
```

Options

This command takes no options.

Arguments

This command takes the following argument:

gt	string	Required. A groupType value for conversion.

Return value

This command returns a hexadecimal version of the supplied value followed by the names of any flags that are set in the value.

Examples

```
explain_groupType [get_object_field groupType]
```

This example returns:

```
80000004 DOMAIN_LOCALSECURITY
```

The unseen value returned by get_object_field groupType in this example was -2147483644, which was converted to the hexadecimal value 80000004 and the name of the set flag DOMAIN_LOCALSECURITY.

Related Tcl library commands

The following commands perform actions related to this command:

- explain_trustAttributes converts a trustAttributes value from an Active Directory object into human-readable form.
- explain_trustDirection converts a trustDirection value from an Active Directory object into human-readable form.
- explain_userAccountControl converts a userAccountControl value from an Active Directory object into human-readable form.

explain_ptype

Use the explain_ptype command to translate the account type for a role assignment into a descriptive text string.

Syntax

```
explain_ptype pt
```

Options

This command takes no options.

Arguments

This command takes the following argument:

pt	string	Required. Specifies the ptype value returned for a role assignment that you want to convert to a text string.

Return value

This command returns a text string that describes the type of account associated with a role assignment.

Examples

```
select_role_assignment "lulu@acme.test/UNIX Login"
get_role_assignment_field ptype
a
explain_ptype a
```

This example returns:

```
All AD users
```

The following table summarizes the descriptive names for different account types that can be associated with a role assignment:

Local UNIX user	#
Local UNIX group	%
Local Windows User	$
Local Windows Group	:
All AD users	a
All Unix users	x
All Windows users	w

explain_trustAttributes

Use the explain_trustAttributes command to convert a trustAttributes value from an Active Directory object field into human-readable form.

Syntax

```
explain_trustAttributes ta
```

Options

This command takes no options.

Arguments

This command takes the following argument:

ta	string	Required. A trustAttributes value for conversion.

Return value

This command returns a hexadecimal version of the supplied value followed by the names of any flags that are set in the value.

Examples

```
explain_trustAttributes [get_object_field trustAttributes]
```

This example returns:

```
8 FOREST_TRANSITIVE
```

The unseen value returned by get_object_field trustAttributes in this example was 8, which was converted to the hexadecimal value 8 and the name of the set flag DOMAIN_LOCALSECURITY.

Related Tcl library commands

The following commands perform actions related to this command:

- explain_groupType converts a groupType value from an Active Directory object into human-readable form.
- explain_trustDirection converts a trustDirection value from an Active Directory object into human-readable form.
- explain_userAccountControl converts a userAccountControl value from an Active Directory object into human-readable form.

explain_trustDirection

Use the explain_trustDirection command to convert a trustDirection value from an Active Directory object field into human-readable form.

Syntax

```
explain_trustDirection td
```

Options

This command takes no options.

Arguments

This command takes the following argument:

td	string	Required. A trustDirection value for conversion.

Return value

This command returns the English version of the trust direction specified by the trustDirection value.

Examples

```
explain_trustDirection [get_object_field trustDirection]
```

This example returns:

two-way

Related Tcl library commands

The following commands perform actions related to this command:

- explain_groupType converts a groupType value from an Active Directory object into human-readable form.
- explain_trustAttributes converts a trustAttributes value from an Active Directory object into human-readable form.
- explain_userAccountControl converts a userAccountControl value from an Active Directory object into human-readable form.

explain_userAccountControl

Use the explain_userAccountControl command to convert a userAccountControl value from an Active Directory object field into a human-readable form.

Syntax

```
explain_userAccountControl uac
```

Options

This command takes no options.

Arguments

This command takes the following argument:

uac	string	Required. A userAccountControl value for conversion.

Return value

This command returns a hexadecimal version of the supplied value followed by the names of any flags that are set in the value.

Examples

```
explain_userAccountControl [get_object_field userAccountControl]
```

returns:

```
10200 ADS_UF_NORMAL_ACCOUNT ADS_UF_DONT_EXPIRE_PASSWD
```

The unseen value returned by get_object_field userAccountControl in this example was 66048, which was converted to the hexadecimal value 10200 and the name of the set flags ADS_UF_NORMAL_ACCOUNT and ADS_UF_DONT_EXPIRE_PASSWD.

Related Tcl library commands

The following commands perform actions related to this command:

- explain_groupType converts a groupType value from an Active Directory object into human-readable form.
- explain_trustAttributes converts a trustAttributes value from an Active Directory object into human-readable form.
- explain_trustDirection converts a trustDirection value from an Active Directory object into human-readable form.

get_all_zone_users

Use the get_all_zone_users command to check Active Directory and return a list of zone users defined within the specified zone and all of its parent zones. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed.

Note that this command does *not* use the currently selected zone to find its list of users. It uses instead the zone specified as an argument for the command. It ignores the currently selected zone. The selected zone remains the selected zone after the command executes.

Syntax

```
get_all_zone_users [-upn] zone_DN
```

Abbreviation

None.

Options

This command takes the following option:

-upn	string	Return user names in the Tcl list as universal principal names (UPNs).

Arguments

This command takes the following argument:

zone_DN	string	Required. The distinguished name (DN) of the zone for which to return users.

Return value

This command returns a Tcl list of zone users defined in the currently selected zone and all of its parent zones. Each entry in the list is in the format sAMAccountName@domain. If a zone user is an orphan user (its corresponding Active Directory user no longer exists), the user is listed by its security identifier (SID) instead of the sAMAccountName.

If the -upn option is present, each entry in the returned Tcl list is a universal principal name (UPN).

Examples

```
get_all_zone_users engineering
```

The example returns the list of zone users:

```
adam.avery@acme.com
brenda.butler@acme.com
chris.carter@acme.com
dave.douglas@acme.com
elliot.evans@acme.com
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_user creates a new zone user and user profile based on a specified Active Directory user.
- create_group creates a new zone group and group profile based on a specified Active Directory group.
- get_effective_groups returns a Tcl list of groups to which a specified user belongs.

get_effective_groups

Use the get_effective_groups command to return the list of effective groups from current zone up the zone hierarchy. Only groups who have a complete profile—whether defined in the current zone or inherited from a parent zone—are included.

The command supports hierarchical zone and classic zones. For classic zones, the command starts from current zone. For hierarchical zones, you can start the search for effective groups at the computer level by specifying the -hostname option.

You can use the adinfo command to return the computer name.

Syntax

```
get_effective_groups [-hostname computer_name]
```

Options

This command takes the following option:

-hostname	string	Specifies the name of the computer to start the search at the computer or computer role level if you run the command in a hierarchical zone with computer-level overrides or computer roles. If you don't specify this option, the search starts in the current zone and computer roles are ignored.

Return value

This command returns a Tcl list of groups with complete profiles in the currently selected zone and all of its parent zones.

Example

```
get_effective_groups -hostname centos7.ajax.com
```

The example returns the list of effective groups starting at the computer level for the computer named centos7.ajax.com.

get_effective_users

Use the get_effective_users command to return the list of effective users from current zone up the zone hierarchy. Only users who have a complete profile—whether defined in the current zone or inherited from a parent zone—are included. Similarly, only users who have a role assignment in the current zone or inherited from a parent zone are included.

The command supports hierarchical zone and classic zones. For classic zones, the command starts from current zone. For hierarchical zones, you can start

the search for effective users at the computer level by specifying the -hostname option.

Syntax

```
get_effective_users [-hostname computer_name]
```

Options

This command takes the following option:

-hostname	string	Specifies the name of the computer to start the search at the computer or computer role level if you run the command in a hierarchical zone with computer-level overrides or computer roles. If you don't specify this option, the search starts in the current zone and computer roles are ignored.

Return value

This command returns a Tcl list of users with complete profiles and at least one role assignment in the currently selected zone and all of its parent zones.

Example

```
get_effective_users -hostname centos7.ajax.com
```

The example returns the list of effective users starting at the computer level for the computer named centos7.ajax.com.

get_user_groups

Use the get_user_groups command to check Active Directory for a specified user and return a list of the groups to which the user belongs. If executed in a script, this command does not output its list to stdout, and no output appears in the shell where the script is executed.

Syntax

```
get_user_groups [-dn] [-z] user_DN|user_UPN
```

Abbreviation

None.

Options

This command takes the following options:

-dn	Return groups in the Tcl list as distinguished names (DNs) instead of user principal names (UPNs).
-z	Restricts the Tcl list of groups to groups that belong to the current zone.

Arguments

This command takes the following argument:

user_DN|user_UPN	string	Required. The user whose groups to return. This argument may specify the user with a distinguished name (DN) or a user principal name (UPN).

Return value

This command used without options returns a Tcl list of all groups listed in Active Directory to which the specified user belongs. Each entry in the list is the user principal name (UPN) of a group that you can use to look up that group.

If the -dn option is set, the Tcl list uses distinguished names (DNs) for groups.

If the -z option is set, the Tcl list is restricted to groups that belong to the currently selected zone.

Note that the command will not return groups for domains that aren't currently bound to ADEdit. If the command finds one or more groups outside of the currently bound domains, it will return a "no binding" message for each unbound domain in which it finds a user's group.

Examples

```
get_user_groups fred.forth@acme.com
```

This example returns a list of groups:

```
poweradmins@acme.com auditors@acme.com
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_group creates a new zone group and group profile based on a specified Active Directory group.
- create_user creates a new zone user and user profile based on a specified Active Directory user.
- get_all_zone_users returns a Tcl list of zone users for the specified zone and all of its parent zones.

get_user_role_assignments

Use the get_user_role_assignments command to retrieve all of the role assignments in the current zone for a specified user. This command returns all of the role assignments from the groups to which the user belongs and the role assignments assigned directly to the user account.

The command checks Active Directory for the user you specify, identifies the groups that the user is a member of, then returns all the role assignments for the list of groups the user is a member and that have been specifically assigned to the user account, including any user role assignments made in computer roles for the currently selected zone.

Syntax

```
get_user_role_assignments [-visible] [-hostname hostname] user_DN
```

Abbreviation

None.

Options

This command takes the following option:

-visible	Specifies that you want to return only visible role assignments in the zone. Use this option to return role assignments for the roles that are identified as visible. This option is only applicable in hierarchical zones.
-hostname	Specifies the computer name to search for role assignments to the user in computer roles. If you set this option, the command checks for computer role assignments in the currently selected zone.

Arguments

This command takes the following argument:

user_DN	string	Required. Specifies the user whose role assignments you want to return. You can use this argument to specify the distinguished name (DN) for a user or a group.

Return value

This command returns a list of all role assignments for the specified Active Directory user in the currently selected zone.

Note that the command does not return role assignments for all zones where the user might be assigned a role.

Examples

```
select_zone
“cn=northamerica,cn=zones,ou=acme,dc=pistolas,dc=org”

get_user_role_assignments
“cn=amy.adams,cn=users,dc=pistolas,dc=org”
```

This example returns a list of groups:

```
{amy.adams@pistolas.org/UNIX Login/northamerica}
{admsf@pistolas.org/Root/sanfrancisco}
{apps@pistolas.org/demos/seattle}
```

Related Tcl library commands

The following commands perform actions related to this command:

- get_all_zone_users returns a Tcl list of zone users for the specified zone and all of its parent zones.
- get_effective_groups returns a list of the groups to which the user belongs.

list_zones

Use the list_zones command to list the zones within a specified domain along with information about each zone. If executed in a script, this command outputs its list to stdout so that the output appears in the shell where the script is executed. The command does not return a Tcl list back to the executing script. Use the ADEdit command get_zones to return a Tcl list.

Syntax

```
list_zones domain
```

Options

This command takes no options.

Arguments

This command takes the following argument:

domain	string	Required. The name of the domain in which to list zones.

Return value

This command returns a list to stdout of the zones within the specified domain. Each entry in the list contains:

- `The zone's distinguished name (DN)
- `The zone type: tree (supported in Server Suite 2012 or later), classic3 or classic4
- `The schema used in the zone

Each entry component is separated from the next by a colon (:).

Examples

list_zones

This example returns a list of zones similar to this:

```
{CN=default,CN=Zones,CN=Acme,DC=acme,DC=com} : classic4 : std
{CN=cz1,CN=Zones,CN=Acme,DC=acme,DC=com} : tree : std
{CN=cz2,CN=Zones,CN=Acme,DC=acme,DC=com} : tree : std
{CN=global,CN=Zones,CN=Acme,DC=acme,DC=com} : tree : rfc
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_assignment creates a new role assignment and saves it to Active Directory.
- precreate_computer creates a zone profile and, if necessary, a new Active Directory computer account.

lmerge

Use the lmerge command to merge and sort the specified lists. You specify the lists to merge as arguments. You must enclose the list commands you want to merge in square brackets.

Syntax

lmerge [list1] [list2] [list[...]]

Options

This command takes no options.

Arguments

This command takes the following arguments:

[list1]	string	Specifies the list command that return the information you want to include first in the merged results.
[list2)	string	Specifies the list command that return the information you want to include second in the merged results.
[list[...]]	string	Specifies any additional list commands that return information you want to include in the merged results.

Return value

This command returns nothing if it runs successfully.

Examples

lmerge [list_zone_users] [list_zone_computers] [list_roles]

This example returns a merged list of zone users, zone computers, and zone roles similar to this:

```
fred@pistolas.org:fred:580398:648:%{u:displayName}:%{home}/%{user}:%{shell}:
lane@pistolas.org:lane:580397:648:%{u:displayName}:%{home}/%{user}:%{shell}:
maya@pistolas.org:maya:580320:648:%{u:displayName}:%{home}/%{user}:%{shell}:
ubu1$@pistolas.org: cpus(1) agentVersion(CentrifyDC 5.2.0): ubu1.pistolas.org
nic3$@pistolas.org: cpus(2) agentVersion(CentrifyDC 5.2.0): nic3.pistolas.org
Rescue - always permit login
listed
UNIX Login
UnixAdminRights
Windows Login
```

You can specify the list arguments using full command names or abbreviations. For example:

```
lmerge [lszc] [lspa]
ubu1$@pistolas.org: cpus(1) agentVersion(CentrifyDC 5.2.0): ubu1.pistolas.org
nic3$@pistolas.org: cpus(2) agentVersion(CentrifyDC 5.2.0): nic3.pistolas.org
dzssh-all/Headquarters : dzssh-* : All of ssh services
login-all/Headquarters : * : Predefined global PAM permission. Do not delete.
```

Related Tcl library commands

None.

modify_timebox

Use the modify_timebox command to modify a timebox value that defines the hours of a week when a role is enabled or disabled. The command defines an hour of the week and then enables or disables that hour in the timebox value. This command is very useful in the set_role_field ADEdit command when setting the timebox field.

Execute this command multiple times on a timebox value to set more than one hour in the value.

For more information about the timebox value format, read the **Timebox Value Format**.

Syntax

```
modify_timebox strTimeBox day hour avail
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

strTimeBox	hex	A 42-digit hexadecimal timebox value. A value of zero disables all hours of the week. A value of FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF enables all hours of the week.
day	integer	Required. The day of the week when the hour occurs. 0=Sunday, 1=Monday, and so on to 6=Saturday.
hour	integer	Required. The hour of the day to enable or disable. Takes a value from 0 to 23. 0 is from midnight to 1 AM, 1 is from 1 AM to 2 AM, and so on to 23, which is from 11 PM to midnight.
avail	integer	Required. Whether to enable or disable the specified hour. 0=disable; all other values=enable.

Return value

This command returns a hexadecimal value that is the timebox value after enabling or disabling the specified hour of the week.

Examples

```
set tb 000000000000000000000000000000000000000000
set tb [modify_timebox $tb 6 23 1]
```

This example returns the modified timebox value:

```
800000000000000000000000000000000000000000
```

Related Tcl library commands

The following commands perform actions related to this command:

- decode_timebox converts an internal timebox value to a decipherable format.

- encode_timebox converts a readable timebox value to an internal timebox format.

precreate_computer

Use the precreate_computer command to create a zone profile for a computer in Active Directory before using the adjoin command to join the domain. The zone profile—a serviceConnectionPoint (scp) object—is usually created by the adjoin command when a computer joins the domain. In some cases, however, creating the zone profile before joining is useful. For example, preparing the computer object before joining enables you to check that you have user profiles and role assignments correctly defined before you join UNIX computers to zones. Verifying this information before the join operation helps to ensure a smooth migration without disrupting users' access to files or applications.

The zone profile is part of an Active Directory computer object. If an Active Directory computer object doesn't exist, precreate_computer can create one and then add the zone profile to the new Active Directory computer object. The zone profile is created in ADEdit's currently selected zone. You can also use the precreate_computer command to specify a container where Active Directory will store the new Active Directory computer object.

You can use the precreate_computer command to create a service connection point for a new or existing Active Directory computer object. You can also use the command to create a computer-specific zone for machine-level zone overrides (in essence a one-computer zone) for the precreated computer. You should note that performing these tasks requires access to the global catalog by default. You can intentionally skip the global catalog search if you know the service connection point you are creating is unique in the forest. However, skipping the global catalog search might prevent you from joining the computer to the domain if there is a conflict.

The precreate_computer command also sets the Active Directory computer object's password and permissions when creating a zone profile. The password is the computer's host name in lower case. The permissions the computer object has are:

- Read and Write permissions to the operatingSystemServicePack, operatingSystem, and operatingVersion attributes of the computer object.
- Read permission for the userAccountControl attribute of the computer object.
- Validate write to the servicePrincipalName and dNSHostName attributes.

You can use precreate_computer to specify a DNS name for the precreated computer and one or more trustees for the precreated computer. Each trustee can be either a user or a group, and has the rights needed to join the computer to the precreated computer account using adjoin.

Use the precreate_computer command option, enctype, to specify encryption types.

The precreate_computer command is similar to using adjoin -precreate, but provides more options and flexibility. You can also precreate computer accounts using Access Manager. For more information about precreating computer accounts, See the **Administrator's Guide for Linux and UNIX**.

Syntax

```
precreate_computer *samaccount@domain*[-ad] [-scp] [-czone] [-all] [-container *rdn*]
[-dnsname *dnsname*] [licensetype *type*] [-trustee *upn*[-trustee *upn*] ...] [nogc]
[stype *spn* [-stype *spn*] ...] [-enctype type [-enctype type] ...]
[-notdelegateanyright]
```

Options

This command takes the following options:

-ad	Creates an Active Directory computer object. precreate_computer won't create an Active Directory computer object if it already exists for the computer specified by the argument upn. Note that if no options specify Active Directory computer object creation and no Active Directory computer object already exists, precreate_computer will fail.
-all	Creates an Active Directory computer object (if one doesn't exist already), a service connection point for the computer object, and a computer zone for the computer object: in essence all of the previous three options combined.
-container	Stores the new Active Directory computer object (if created) in the Active Directory container specified by rdn, which is the relative distinguished name (RDN) of the container. The root of the specified Active Directory container is the distinguished name (DN) of the current domain. precreate_computer appends the RDN to the root DN to come up with the container DN.
-czone	Creates a computer zone for the computer object.
	Sets the DNS name for the computer account to the provided DNS name. If this option isn't present, the precreate_computer

-dnsname	command automatically sets the DNS name for the computer account. It derives the DNS name from the computer's sAMAccount name and the domain name.
-enctype	Set the msDS encryption types permitted in precreate _computer command. Default is 31. Options are: aes256-cts-hmac-sha1-96, aes256-cts aes128-cts-hmac-sha1-96, aes128-cts arcfour-hmac, rc4-hmac, arcfour-hmac-md5 des-cbc-md5, des des-cbc-crc
-licensetype	Specifies the type of license a computer uses. The valid values are server workstation
-nogc	Allows you to create the computer account without binding to a global catalog domain controller. You should only use this option if you know the computer scp object does not exist in the domain.
-notdelegateanyright	Allows you to create the computer account without delegating any rights. If you specify this option, note that the -trustee option has no effect.
-scp	Creates a service connection point for the Active Directory computer object.
-stype	Specifies the service principal types to create for a precreated computer account. You can specify multiple -stype options, with each specifying a different service principal type. If you don't specify this option, the precreate_computer command automatically creates the several default service principal names for the following service principal types: ipp afpserver nfs cifs ftp http host For each type of service, the precreate_computer command specifies two service principal names in the form of *serviceNamel computerName* and *serviceName/computerName.domain.com*. For example: ftp/rhel6 ftp/rhel6.acme.com If you specify one or more -stype options, only the service principal names for those service types are created for the precreated computer account.
-trustee	Gives the user or group specified by the *upn* argument permission to join a computer to the precreated computer account. You can specify multiple -trustee options, with each specifying a different user or group, to give multiple users and groups permission to join a precreated computer to a zone.

Arguments

This command takes the following argument:

samaccount@domain	string	Required. Specifies the name of the computer account and the domain to join. The computer name is the sAMAccountName for the account in the form of *computer$*. For example: engserv$@acme.com

Return value

This command returns nothing if it runs successfully.

Examples

```
precreate_computer redhat\$@acme.com -trustee adam.avery@acme.com
\-trustee martin.moore@acme.com -enctype arcfour-hmac
```

This example precreates a zone profile in the currently selected zone for the computer "redhat$@acme.com", and specifies as trustees the Active Directory users Adam Avery and Martin Moore.

Because the example does not include the -stype option, this example also automatically creates the following default service principal names for services on the computer:

- ipp/redhat and ipp/redhat.acme.com
- afpserver/redhat and afpserver/redhat.acme.com
- nfs/redhat and nfs/redhat.acme.com
- cifs/redhat and cifs/redhat.acme.com
- ftp/redhat and ftp/redhat.acme.com

- http/redhat and http/redhat.acme.com
- host/redhat and host/redhat.acme.com

Related Tcl library commands

The following commands perform actions related to this command:

- list_zones returns a list of zones in a specified domain to stdout.
- create_assignment creates a new role assignment and saves it to Active Directory.

remove_user_from_group

Use the remove_user_from_group command to remove an Active Directory user from an Active Directory group.

Syntax

```
remove_user_from_group user group
```

Options

This command takes no options.

Arguments

This command takes the following arguments:

user	string	Required. The user principal name (UPN) of the Active Directory user to remove.
group	string	Required. The UPN of the Active Directory group from which to remove the user.

Return value

This command returns nothing if it runs successfully.

Examples

```
remove_user_from_group adam.avery@acme.com pubs@acme.com
```

Related Tcl library commands

The following commands perform actions related to this command:

- create_aduser creates a new Active Directory user account and sets its password.
- create_adgroup creates a new Active Directory group account and specifies its scope.
- create_user creates a new zone user and user profile based on an existing Active Directory user.
- create_group creates a new zone group and group profile based on an existing Active Directory group.
- add_user_to_group adds an Active Directory user to an Active Directory group.

set_change_pwd_allowed

Use the set_change_pwd_allowed command to modify the ADS_UF_PASSWD_CANT_CHANGE flag in the UserAccountControl attribute. This flag controls whether an Active Directory user can change his or her domain password. You must specify the distinguished name of a valid Active Directory user account that should be allowed to change his or her password.

Syntax

```
set_change_pwd_allowed userdn
```

Options

This command takes no options.

Arguments

This command takes the following argument:

userdn	string	Required. Specifies the distinguished name of the Active Directory user who is allowed to change his or her password.

Return value

This command returns nothing if it runs successfully.

Examples

```
set_change_pwd_allowed
CN=frank.smith,CN=Users,DC=ajax,DC=test

get_object_field sd

(OA;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;WD)
(OA;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;PS)
```

This example deselects the "User cannot change password" account property for the Active Directory user frank.smith.

Related Tcl library commands

The following commands perform actions related to this command:

- create_aduser creates a new Active Directory user account and sets the password for the account.
- set_change_pwd_denied prevents an Active Directory user from changing the domain password for his or her account.

set_change_pwd_denied

Use the set_change_pwd_denied command to modify the ADS_UF_PASSWD_CANT_CHANGE flag in the UserAccountControl attribute. This flag controls whether an Active Directory user can change his or her domain password. You must specify the distinguished name of a valid Active Directory user account that should not be allowed to change his or her password.

Syntax

```
set_change_pwd_denied userdn
```

Options

This command takes no options.

Arguments

This command takes the following argument:

userdn	string	Required. Specifies the distinguished name of the Active Directory user who is not allowed to change his or her password.

Return value

This command returns nothing if it runs successfully.

Examples

```
set_change_pwd_denied CN=adam.avery,CN=Users,DC=ajax,DC=test

get_object_field sd

(OD;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;WD)
(OD;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;PS)
```

This example selects the “User cannot change password” account property for the Active Directory user adam.avery.

Related Tcl library commands

The following commands perform actions related to this command:

- create_aduser creates a new Active Directory user account and sets the password for the account.
- set_change_pwd_allowed allows an Active Directory user to change the domain password for his or her account.

Timebox Value Format

A Delinearole specifies a collection of rights. A role object contains a field, timebox, that defines what hours in a week a role is either enabled or disabled. Setting the timebox field in a role object defines when a role's rights are in effect.

You can read a role's timebox field using the ADEdit command get_role_field and set the timebox value using set_role_field. You can modify an existing timebox value one hour at a time using the ADEdit library command modify_timebox.

To interpret a timebox value, or to set it directly, you must know the timebox value format which is, unfortunately, not simple as defined by Active Directory. This appendix explains the format.

Hex string

The timebox value is a 42-character (21-byte) hexadecimal value stored as a string. When the hex value is converted to a binary value, its 168 bits each map to a single hour within the week. If a bit is set to 1, its corresponding hour is enabled for the role. If set to 0, its corresponding hour is disabled.

After you define the 168 bits using a hexadecimal value, you can usee the encode_timebox function to convert the value into an internal format that specifies when a role is available to use.

Hour mapping

Each day of the week takes three bytes (24 bits) to specify how its hours are enabled or disabled. The following tables show how the hours of a day are mapped to the bits within each of a day's three bytes.

Byte 0

12-1 AM	0 (least-significant bit)
1-2 AM	1
2-3 AM	2
3-4 AM	3
4-5 AM	4
5-6 AM	5
6-7 AM	6
7-8 AM	7 (most-significant bit)

Byte 1

8-9 AM	0 (least-significant bit)
9-10 AM	1
10-11 AM	2
11-12 AM	3
12-1 PM	4

1-2 PM	5
2-3 PM	6
3-4 PM	7 (most-significant bit)

Byte 2

4-5 PM	0 (least-significant bit)
5-6 PM	1
6-7 PM	2
7-8 PM	3
8-9 PM	4
9-10 PM	5
10-11 PM	6
11-12 PM	7 (most-significant bit)

Day mapping

Each of the seven days in a week have three bytes within the 21-byte timebox value. These bytes are in chronological order from most-significant byte to least-significant byte. (Note that this is the opposite of chronological bit order within each byte, which is LSB to MSB.) The starting point of a week is 4 PM on Saturday afternoon.

The table below shows how each day's three bytes (0-2) map to the timebox value's bytes, listed here in order from most-significant byte to least-significant byte.

Saturday, byte 2	20 (most-significant byte)
Sunday, byte 0	19
Sunday, byte 1	18
Sunday, byte 2	17
Monday, byte 0	16
Monday, byte 1	15
Monday, byte 2	14
Tuesday, byte 0	13
Tuesday, byte 1	12

Tuesday, byte 2	11
Wednesday, byte 0	10
Wednesday, byte 1	9
Wednesday, byte 2	8
Thursday, byte 0	7
Thursday, byte 1	6
Thursday, byte 2	5
Friday, byte 0	4
Friday, byte 1	3
Friday, byte 2	2
Saturday, byte 0	1
Saturday, byte 1	0 (least-significant byte)

Using ADEdit with Classic Zones

Delineasupports both classic and hierarchical zones. If you have upgraded agents to a version of Delineasoftware that supports hierarchical zones (version 5.x or later), you can choose to either migrate your classic zones into a hierarchical zone structure or maintain them as classic zones.

If you choose to maintain any zones as classic zones, however, you should be aware that the authorization model in classic zones differs from the authorization model used in hierarchical zones. For example, in classic zones, authorization is an optional feature that can be enabled or disabled. If authorization is not enabled in a classic zone, any user with a valid profile in a zone is automatically granted login access to all computers joined to that zone.

Because authorization is handled differently in classic zones, there are specialized ADEdit commands and command options for creating and managing rights and roles in classic zones. The commands in this appendix are only applicable when you are working with classic zones.

Enabling Authorization in Classic Zones

The following ADEdit commands are used to enable or disable authorization in a classic zone and to check whether authorization is currently enabled or disabled.

is_dz_enabled	Checks whether authorization is enabled in a currently selected classic zone.
manage_dz	Enables or disables authorization in classic zones.

Working with privileged Commands and PAM Applications

With some limitations, you can use most of the ADEdit commands for working with rights, role definitions, and role assignments in classic zones in the same way you work with them in hierarchical zones. In a classic zone, however, you must explicitly enable authorization for the zone. Thereafter, defining rights and roles or making role assignments work the same in classic zones and hierarchical zones.

In most cases, any differences or limitations for classic zones involve options or arguments that are not supported or not applicable in classic zones. For example, fields such as allowLocalUser, alwaysPermitLogin, and auditLevel are not applicable in classic zones. You can use the set_role_field command to set other field values in a classic zone. Individual commands specify these types of limitations.

Working with Restricted Shell Environments and Commands

Before you can use the restricted shell (dzsh) to run commands in a classic zone, you must create the restricted shell environment. After you have created the restricted shell environment in your working context, you can run restricted shell commands in that dzsh context.

Restricted commands cannot be assigned to a role directly. A restricted shell environment has to be created first. The restricted shell commands can then be created under the currently selected restricted shell environment. Only one restricted shell environment can be assigned to a role. The restricted shell environment and privileged UNIX commands cannot be assigned to a role simultaneously. Assigning a new restricted shell environment to a role removes all of the previously defined privileged UNIX commands from the restricted shell. Assigning new privileged commands to a role that previously had a restricted shell environment removes the restricted shell environment and any restricted shell commands defined for the restricted shell environment.

Setting up the restricted shell environment

The following ADEdit commands are used to set up and manage the restricted shell environment prior to working with any restricted shell commands.

clear_rs_env_from_role	Removes the restricted shell environment from the currently selected role that is stored in memory.
delete_rs_env	Deletes the currently selected restricted environment from Active Directory and also from memory.
get_role_rs_env	Gets the restricted shell environment from the currently selected role that is stored in memory.
get_rs_envs	Gets the list of restricted environments that are defined within the currently selected zone.

get_rse_cmds	Gets a Tcl list of restricted shell commands associated with the currently selected restricted shell environment.
get_rse_field	Gets the value for a specified field from the restricted shell environment stored that is stored in memory.
list_rs_envs	Prints a list of restricted shell environments defined for the currently selected zone to stdout.
new_rs_env	Creates a new restricted shell environment for the current zone, stores it in memory, and sets it to be the currently selected restricted shell environment.
save_rs_env	Saves the currently selected restricted environment that is stored in memory to Active Directory.
select_rs_env	Retrieves a restricted shell environment for the currently selected zone from Active Directory, stores it in memory, and sets it to be the currently selected restricted shell environment for other ADEdit commands.
set_rs_env_for_role	Assigns a restricted shell environment to the currently selected role that is stored in memory.
set_rse_field	Sets the value for a specified field in the currently selected restricted shell environment stored in memory.

Using restricted commands

The following ADEdit commands are used to set up and manage the restricted shell restricted shell commands.

delete_rs_command	Deletes the currently selected restricted shell command from Active Directory and from memory.
get_role_rs_commands	Returns a Tcl list of restricted shell commands associated with the currently selected role.
get_rs_commands	Checks Active Directory and returns a Tcl list of restricted shell commands defined for the currently selected zone.
get_rsc_field	Gets the value for a specified field from the currently selected restricted shell command that is stored in memory.
list_rs_commands	Prints a list of restricted shell commands defined for the currently selected zone to stdout.
new_rs_command	Creates a new restricted shell command under the currently selected restricted shell environment, stores it in memory, and sets it to be the currently selected restricted shell command.
save_rs_command	Saves the currently selected restricted shell command that is stored in memory to Active Directory.
select_rs_command	Retrieves a restricted shell command in the currently selected zone from Active Directory, stores it in memory, and sets it to be the currently selected restricted shell command for other ADEdit commands.
set_rsc_field	Sets the value for a specified field for the currently selected restricted shell command that is stored in memory.

Creating computer-level role assignments in classic zones

Classic zones support computer-level role assignments. If you want to configure computerlevel role assignments, keep the following in mind:

- The classic zone that the computer is a member of must have authorization enabled before you can create role definitions and role assignments.
- The role assignment is only valid on the computer where you have made the assignment.
- The role definition you use must be defined in the classic zone that the computer is a member of.

A computer-level role assignment in a classic zone is similar to computer-level overrides in hierarchical zones, except that you cannot save user or group profile information for individual computers. User and group information is stored in the classic zone. To enable computer-specific role assignments in classic zones, you must use a specialized zone type, the classic-computer zone type.

To create a computer-level role assignment in a classic zone:

1. Precreate the computer in a classic4 zone, if it doesn't already exist.
2. Create a zone that uses the specialized zone type of classic-computer.
3. Select the classic-computer zone within the classic zone.
4. Create the role assignment.

The following code snippet illustrates the commands to execute in ADEdit to make computer-specific role assignments in classic zones:

```
bind ajuba.net
package require ade_lib
    1.0
select_zone cn=cls,cn=zones,dc=ajuba,dc=net
get_zone_field type
    classic4
precreate_computer rhelqa$@ajuba.net
get_zone_computers
    {comp5$@ajuba.net} {rhelqa$@ajuba.net}
create_zone classic-computer rhelqa.ajuba.net@cn=cls,cn=zones,dc=ajuba,dc=net
select_zone rhelqa.ajuba.net@cn=cls,cn=zones,dc=ajuba,dc=net
new_role_assignment user5@ajuba.net
set_role_assignment_field role role1/cls
save_role_assignment
```

You can then get the classic-computer zones by running the get_child_zones command when the classic zone is selected. For example:

```
select_zone cn=cls,cn=zones,dc=ajuba,dc=net
get_child_zones
helqa.ajuba.net@CN=c122,CN=Zones,DC=ajuba,DC=net
comp5.ajuba.net@CN=c122,CN=Zones,DC=ajuba,DC=net
```

Quick reference for commands and library procedures

The following table lists the ADEdit and ade_lib commands in alphabetical order. The table summarizes the command syntax for each command with optional elements in [square brackets] and variables in italics. For more detailed information about any command, see the previous sections **ADEdit command reference** or **ADEdit Tcl Procedure Library Reference**.

add_command_to_role command[/ zonename]	acr	
add_map_entry key value	ame	
add_map_entry_with_comment key value comment	amewc	
add_object_value dn field value	aov	
add_pamapp_to_role app[/zonename]	apr	
add_sd_ace sddl_string ace_string	ase	
add_user_to_group user group		X
bind [-gc] [-write] [-machine] [server@]domain [user [password]]		
clear_rs_env_from_role	crse	
convert_msdate msdate		X
create_adgroup dn sam gtype		X
create_aduser dn upn sam pw		X
create_assignment upn role [/zonename] [from] [to] [description]		X
create_computer_role computer_role_path group_upn	ccr	
create_dz_command name command description form dzdo_runas dzsh_runas flags pri umask path		X
create_group upn name gid		X
create_nismap map key:value comment		X
create_pam_app name application description		X
create_role name description sysrights pamrights cmdrights allowlocal rsenv visible		X
create_rs_command rsc_name cmd description form dzsh_runas flags pri umask path		X
create_rs_env rse_name rse_description		X
create_user ad uname uid gid gecos home shell role		X
create_zone [-ou] zone_type path [schema_type]	cz	
decode_timebox strTimeBox		X
delegate_zone_right right principal_upn		

delete_dz_command	dldzc	
delete_local_group_profile group_name	dllgp	
delete_local_user_profile user_name	dllup	
delete_map_entry key:index	dlme	
delete_nis_map	dlnm	
delete_object	dlo	
delete_pam_app	dlpam	
delete_role	dlr	
delete_role_assignment	dlra	
delete_rs_command	dlrsc	
delete_rs_env	dlrse	
delete_sub_tree dn		
delete_zone	dlz	
delete_zone_computer	dlzc	
delete_zone_group	dlzg	
delete_zone_user	dlzu	
dn_from_domain domain_name	dnfd	
dn_to_principal [-upn] principal_dn	dntp	
domain_from_dn domain_name	dfdn	
encode_timebox strTimeBox		X
explain_groupType gt		X
explain_ptype pt		X
explain_sd sddl_string		
explain_trustAttributes ta		X
explain_trustDirection td		X
explain_userAccountControl uac		X
get_adinfo domain|zone|host	adinfo	
get_all_zone_users [-upn] zone_DN		X

get_bind_info domain forest|server|sid|domain_level|forest_level	gbi	
get_child_zones [-tree] [-crole] [-computer]	gcz	
get_dz_commands	gdzc	
get_dzc_field field	gdzcf	
get_effective_groups [-hostname computer_name]		X
get_effective_users [-hostname computer_name]		X
get_group_members [-ad | -upn] group_UPN	ggm	
get_local_group_profile_field field_name	glgpf	
get_local_groups_profile	glgp	
get_local_user_profile_field field_name	glupf	
get_local_users_profile	glup	
get_nis_map	gnm	
get_nis_map_field field	gnmf	
get_nis_map_with_comment	gnmwc	
get_nis_maps	gnms	
get_object_field field	gof	
get_object_field_names	gofn	
get_objects [-gc] [-depth one|sub] [-limit limit] [-f forest] base filter	go	
get_pam_apps	gpam	
get_pam_field	gpf	
get_parent_dn DN	gpd	
get_pending_zone_groups	gpzg	
get_pending_zone_users	gpzu	
get_pwnam unix_name	gpn	
get_rdn DN	grdn	
get_role_apps	grap	
get_role_assignment_field field	graf	
get_role_assignments [-upn]	gra	

get_role_commands	grc	
get_role_field field	grf	
get_role_rs_commands	grrsc	
get_role_rs_env	grrse	
get_roles	getr	
get_rs_commands	grsc	
get_rs_envs	grse	
get_rsc_field field	grscf	
get_rse_cmds	grsec	
get_rse_field field	grsef	
get_effective_groups [-dn] [-z] user_DN|user_UPN		X
get_user_role_assignments [-visible] [-hostname hostname] user_DN		X
get_schema_guid schema_name	gsg	
get_zone_computer_field field	gzcf	
get_zone_computers	gzc	
get_zone_field field	gzf	
get_zone_group_field field	gzgf	
get_zone_groups	gzg	
get_zone_nss_vars	gznv	
get_zone_user_field field	gzuf	
get_zone_users [-upn]	gzu	
get_zones domain	gz	
getent_passwd	gep	
guid_to_id guid		
help command_pattern	h	
is_dz_enabled	idze	
joined_get_user_membership user_UPN	jgum	
joined_name_to_principal [-upn] UNIX_name	jntp	

joined_user_in_group user_UPN group_UPN	jug	
list_dz_commands	lsdzc	
list_local_groups_profile	lslgp	
list_local_users_profile	lslup	
list_nis_map	lsnm	
list_nis_map_with_comment	lsnmwc	
list_nis_maps	lsnms	
list_pam_apps	lspa	
list_pending_zone_groups	lpzg	
list_pending_zone_users	lpzu	
list_role_assignments [-upn] [-visible] [-userl-groupl-invalid]	lsra	
list_role_rights	lsrr	
list_roles	lsr	
list_rs_commands	lsrsc	
list_rs_envs	lsrse	
list_zone_computers	lszc	
list_zone_groups	lszg	
list_zone_users [-upn]	lszu	
list_zones domain		X
lmerge [list] [list] [list...]		X
manage_dz -onl-off	mnz	
modify_timebox strTimeBox day hour avail		X
move_object destinationDN	mvo	
new_dz_command name	newdzc	
new_local_group_profile group_name	newlgp	
new_local_user_profile user_name	newlup	
new_nis_map [-automount] map	newnm	
new_object dn	newo	

new_pam_app name	newpam	
new_role name	newr	
new_role_assignment upn	newra	
new_rs_command name	newrsc	
new_rs_env name	newrse	
new_zone_computer sAMAccountName@domain	newzc	
new_zone_group AD_group_UPN	newzg	
new_zone_user AD_user_UPN	newzu	
pop		
precreate_computer AMAccount@domain [-ad] [-scp] [-czone] [-all] [container rdn] [-dnsname dnsname] [-trustee upn [-trustee upn] ...] [-nogc] [stype spn [-stype spn] ...]		X
principal_from_sid [-upn] sid	pfs	
principal_to_dn principal_upn	ptd	
principal_to_id [-apple] upn	pti	
push		
quit	q	
remove_command_from_role command[/zonename]	rcfr	
remove_object_value dn field value	rov	
remove_pamapp_from_role app[/zonename]	rpamfr	
remove_sd_ace sddl_string ace_string	rsa	
remove_user_from_group user group		X
rename_object name	rno	
save_dz_command	svdzc	
save_local_group_profile	svlgp	
save_local_user_profile	svlup	
save_nis_map	svnm	
save_object	svo	
save_pam_app	svpam	
save_role	svr	

save_role_assignment	svra	
save_rs_command	svrsc	
save_rs_env	svrse	
save_zone	svz	
save_zone_computer	svzc	
save_zone_group	svzg	
save_zone_user	svzu	
select_dz_command command	sldzc	
select_local_group_profile roup_name	sllgp	
select_local_user_profile user_name	sllup	
select_nis_map map	slnm	
select_object [-rootside] [-attrs a1[,a2,...]] dn	slo	
select_pam_app name	slpam	
select_role role	slr	
select_role_assignment principal/role [/zone]	slra	
select_rs_command rs_cmd	slrsc	
select_rs_env rse	slrse	
select_zone path	slz	
select_zone_computer sAMAccountName@domain	slzc	
select_zone_group D_group_UPN	slzg	
select_zone_user user	slzu	
set_change_pwd_allowed userdn		
set_change_pwd_denied userdn		
set_dzc_field field value	sdzcf	
set_ldap_timeout timeout_in_seconds		
set_local_group_profile_field field_name value	slgpf	
set_local_user_profile_field field_name value	slupf	
set_object_field field value	sof	

set_pam_field field value	spf	
set_role_assignment_field field value	sraf	
set_role_field field value	srf	
set_rs_env_for_role environment	srse	
set_rsc_field field value	srscf	
set_rse_field field value	srsef	
set_sd_owner sddl_string owner_sid	sso	
set_user_password principal_UPN password	sup	
set_zone_computer_field field value	szcf	
set_zone_field field value	szf	
set_zone_group_field field value	szgf	
set_zone_user_field field value	szuf	
show [all | bind | zone | user | computer | assignment | object | group | pamright | dzcommand | nismap | role | license | rse | rs_command]		
sid_to_escaped_string sid	stes	
sid_to_uid sid	stu	
validate_license path	vl	

This section contains topics on scripting operations using Microsoft PowerShell for use with the Server Suite Authentication and Privilege Elevation services.

- Scripting Access Control and Privilege Management with PowerShell
- Auditing and Analysis Scripting Guide

Scripting Access Control and Privilege Management with PowerShell

Note: Subject matter (not formatting) last updated December 2021 (release 2021.1).

Introduction

Overview

This topic discusses access control and privilege management using PowerShell-based command-line programs. This information is intended to help you develop scripts for creating and populating zones and performing other administrative tasks on Windows computers. With scripts, you can automate the administrative tasks you might otherwise perform using the Access Manager console.

Specifically, the topic describes the Delinea authentication and privilege PowerShell-based command set. These PowerShell cmdlets run on Windows computers and can be used to automate access control and privilege management tasks, such as the creation of Delinea zones, rights, and roles. You can also use the cmdlets to perform other administrative tasks. For example, you can write scripts to add UNIX profiles for Active Directory users and groups to Delinea zones, assign UNIX and Windows users and groups to roles, and manage network information through NIS maps.

Intended audience

This topic provides information for Active Directory administrators who want to use PowerShell scripts to install or maintain Delinea software. This document supplements the help provided within the PowerShell environment using the get-help function. Whereas the get-help function describes each cmdlet in detail, this document introduces the access module for Windows PowerShell objects and how you can use PowerShell cmdlets and scripts to perform access control and privilege management tasks.

This topic assumes general knowledge of Microsoft Active Directory, PowerShell scripts and syntax, and the Windows PowerShell modules used to write scripts for Active Directory. You should also understand the structure of Active Directory, including the Active Directory schema your organization is using.

In addition to scripting skills, you should be familiar with Delinea architecture, terms, and concepts, and understand how to perform administrative tasks for authentication and privilege elevation and for the UNIX platforms you support.

Subtopics

This topic is divided into these subtopics:

- **Developing Scripts for Administrative Tasks:** An introduction to access control and privilege management using Windows PowerShell.
- **Installing the PowerShell Access Module:** How to download and install the module as a separate package.
- **Managing Delinea Objects using Windows PowerShell Scripts:** How to use cmdlets to connect to Active Directory and perform access control and privilege management tasks.
- **Objects and Properties:** Lists the objects defined by the authentication and privilege-elevation PowerShell module and the properties of each object.
- **Adding Users in a One-way Trust Environment:** How to add a user in a one-way trust environment using the authentication and privilege-elevation PowerShell module.
- **Using Predefined Scripts to Generate Reports:** Describes predefined report scripts that are included with the authentication and privilege-elevation PowerShell module and how to configure report output files to generate HTML- and PDF-formatted report files.

Compatibility and Limitations

The information in this topic is intended for use with Server Suite, version 5.1.x or later and Server Suite 2017.2 or later. Although intended to be accurate and up to date, interfaces are subject to change without notice and can become incompatible or obsolete when a newer version of the software is released.

In general, APIs attempt to be backward-compatible but are not guaranteed to work with older versions of the software. Because the authentication and privilege elevation cmdlets are subject to change, enhancement, or replacement, the information in this topic can also become incomplete, obsolete, or unsupported in future versions. If you are unsure whether this topic is appropriate for your software version, consult the Delinea Web site or Delinea Support to find out if another, more appropriate, topic is available.

Developing Scripts for Administrative Tasks

This section introduces access control and privilege management using Windows PowerShell. It consists of the following:

- APIs in the form of PowerShell command-line programs, called cmdlets, that are packaged in Dynamic Link Libraries (DLLs).
- A PowerShell help file that includes complete cmdlet reference information and this scripting guide.
- Sample scripts to illustrate administrative tasks.
- Predefined scripts to generate reports.
- Individual help files for each predefined report script.

On Windows computers, you can use the authentication and privilege elevation module for Windows PowerShell to develop your own custom scripts that access, create, or modify Delinea-specific data in Active Directory.

Getting Started with cmdlets for PowerShell

The access module for PowerShell consists of cmdlets that you can use to manage Delinea-specific information in Active Directory. A *cmdlet* is a lightweight command-line program that runs in the Windows PowerShell environment. In most cases, cmdlets perform a basic operation and return a Microsoft .NET Framework object to the next command in the pipeline.

The cmdlets in the Delinea module enable you to access, create, modify, and remove information about Delinea zones, including details for each zone about the defined user, group, and computer profiles; all aspects of the rights, roles, and role assignments; and the available NIS maps and NIS map entries. You can combine cmdlets and use them in scripts to automate administrative tasks, such as user or group profile provisioning or creating rights, roles, and role assignments.

In most cases, you can use cmdlets to manipulate Delinea objects in any type of zone. However, because the implementation of authorization differs greatly in hierarchical zones from authorization in classic zones, the access module for Window PowerShell cmdlets that enable you to create and work with rights, roles, or role assignments are only applicable in hierarchical zones. You should not use the cmdlets for rights, roles, and role assignments in classic zones.

Managing UNIX information from a Windows Computer

You can use the cmdlets to work with information for any Delinea-managed computer and to manage UNIX profiles and access rights. However, you can only run the cmdlets on Windows-based computers that have the Windows PowerShell command-line shell available. If you want to develop scripts that run on UNIX computers, you can use the ADEdit program (adedit). The ADEdit application provides functionality similar to the cmdlets. For detailed information about using ADEdit, see the ADEdit Command Reference and Scripting topic.

Writing Programs in Other Languages

If you want to develop programs or scripts that run on Windows but outside of the Windows PowerShell environment, you can use any language that supports the Component Object Model (COM) interface. The Delinea COM-based interface is available as part of the Delinea Windows Software Development Kit (SDK). The SDK package is a completely separate API that provides reusable objects that you can call in programs written in .NET or COM-enabled languages. You can, therefore, create or modify your own applications to use these objects in VBScript and JScript or in .NET-compliant (such as C#) languages. For more information about using the COM-based API, see the Windows API Programmer's Guide.

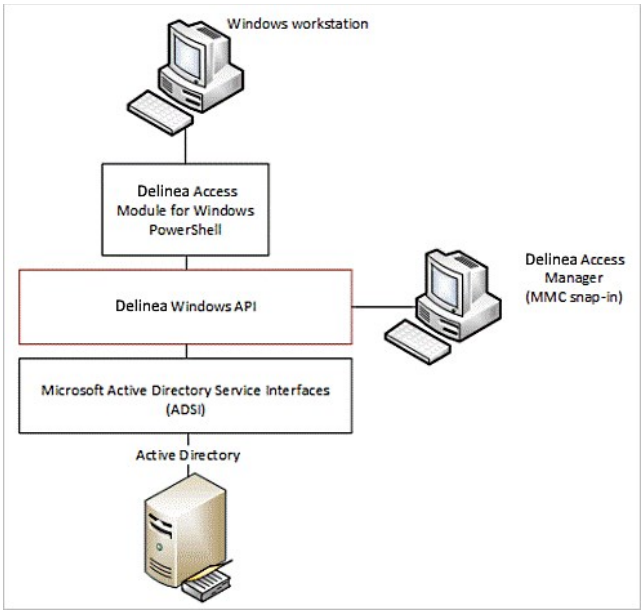
Accessing information stored in Active Directory

The Delinea access module for PowerShell cmdlets connect to Active Directory to access all of the Delinea-specific information stored there. You can, therefore, write PowerShell scripts to automate procedures that you would otherwise have to perform using access manager.

The cmdlets rely on the underlying interfaces provided by Microsoft Active Directory Service Interfaces (ADSI) and the Delinea Windows API. The ADSI layer provides low-level functions that permit applications to read and write data in Active Directory. The cmdlets provide a task and object-based level of abstraction for retrieving and manipulating Delinea-specific information so that you do not need to know the details of how the data is stored or how to use any of the underlying ADSI functions directly.

The following figure illustrates how the Delinea access module for PowerShell provides an abstraction layer between the data stored in Active Directory and your scripting environment.

Figure: PowerShell Abstraction Layer



The Active Directory schema defines how all of the objects and attributes in the database are stored. When you add Delinea objects to the Active Directory database, how that data is stored depends on the Active Directory schema you have installed. The Delinea access module for PowerShell, however, provides a logical view of the data, eliminating the need to know the details of how data is stored in different schemas when performing common administrative tasks. The cmdlets also provide a simple and Delinea-focused method for accessing subject UNIX objects.

Using the cmdlets, you can write scripts that automatically create and manage zones or update user, group, or computer properties. In most cases, the cmdlets enable you to perform exactly the same tasks from the command line that you would otherwise perform interactively using access manager.

Installing the PowerShell Access Module

This section explains how to download and install the module as a separate package. You can install the authentication and privilege elevation module for PowerShell from the Server Suite setup program or as a separate package. This section includes the access control and privilege management cmdlets for Windows PowerShell, sample scripts, and documentation for performing common administrative tasks using PowerShell scripts. This section describes how to install the software if you download it as a separate package or run the package-specific setup program on a Windows computer.

Selecting and Downloading a Standalone Package

The cmdlets that run in Windows PowerShell are defined in DLLs that can be installed on any computer where you install other Windows-based components, such as the Access Manager console. You can also download these libraries separately, along with sample scripts and documentation, onto computers where access manager is not installed.

> **Note:** You can download the access module for PowerShell as a separate package from the Delinea Download Center under Software Development Kits. However, you must obtain an unlocking code or license key from your Delinea sales representative to access the module.

Running the Setup program

After you have downloaded the compressed file to your computer, you can extract the files and run the setup program to install the access module for PowerShell files.

To use the authentication and privilege elevation module for Windows PowerShell on a Windows Server server-core computer, you must have Windows PowerShell, version 2.0 or later, installed first. Also, install the authentication and privilege elevation module for Windows PowerShell on a Windows Server Core environment in silent mode, due to a user interface limitation. Please check the process exit code to see whether the installation succeeded or failed.

> **Note:** Server core is a minimal installation option that is available when you are deploying Windows Server. Server core includes most but not all server roles. Server Core has a smaller attack surface due to a smaller code base.

To run the standalone setup program:

1. Download the file.
2. Right-click downloaded file and select **Extract All** to extract the compressed files to a folder.
3. Double-click the standalone executable to start the setup program. For example, for the 64-bit version of the file, double click the CentrifyDC_PowerShell-5.2.0-win64.exe file.

   Note: Alternatively, you can install from the Microsoft Installer (.msi) file. For example, you might run the following command: msiexec.exe /i "CentrifyDC_PowerShell-5.2.0-win64.msi" /norestart.

   The Welcome page appears.
4. Click the **Next** button. The License Agreement page appears.
5. Click to select the **I accept the terms in the License Agreement** check box.
6. Click the **Next** button. The Location page appears.
7. Accept the default location or click **Change** to choose a different one. If you accept the default location, the authentication and privilege elevation cmdlets are in a separate authentication and privilege elevation for Windows PowerShell console. If you want the authentication and privilege elevation cmdlets to be available in the default Windows PowerShell console with other PowerShell modules, select the following location:

   C:\Windows\System32\WindowsPowerShell\v1.0\Modules\Centrify.DirectControl.PowerShell
8. Click the **Next** button.
9. Click the **Install** button.
10. Click the **Finish** button to complete the installation.

Importing cmdlets into the Windows PowerShell Console

If you install the authentication and privilege elevation module for Windows PowerShell in the default location, it is a self-contained Windows PowerShell console. If you install the files in the location for system modules so that cmdlets from other modules are available in the same console, you should import the authentication and privilege elevation module into your default Windows PowerShell console.

To import the authentication and privilege elevation module:

1. On the **Start** menu, select **Windows PowerShell** to display a menu extension with a list of tasks.
2. On the tasks menu, select **Import System Modules** to import the authentication and privilege elevation module and open the Windows PowerShell console.
3. Verify the installation and import completed successfully by typing the following command at the PowerShell prompt:

   get-command -Cdm

   You should see a listing of the authentication and privilege elevation cmdlets, similar to the following partial list:

```
CommandType   Name                  Definition
-----------   ----                  ----------
 Cmdlet       Add-CdmApplicationRight   Add-CdmApplicationRight -Right ...
 Cmdlet       Add-CdmCommandRight       Add-CdmCommandRight -Right <Cdm...
 Cmdlet       Add-CdmDesktopRight       Add-CdmDesktopRight -Right <Cdm...
 Cmdlet       Add-CdmNetworkAccessRight Add-CdmNetworkAccessRight -Righ...
 Cmdlet       Add-CdmPamRight           Add-CdmPamRight -Right <CdmPamR...
 Cmdlet       Add-CdmSshRight           Add-CdmSshRight -Right <CdmSshR...
 Cmdlet       Get-CdmApplicationRight   Get-CdmApplicationRight [-Zone ...
 Cmdlet       Get-CdmCommandRight       Get-CdmCommandRight [-Zone <Cdm...
 Cmdlet       Get-CdmComputerRole       Get-CdmComputerRole -Zone <CdmZ...
 Cmdlet       Get-CdmDesktopRight       Get-CdmDesktopRight [-Zone <Cdm...
 Cmdlet       Get-CdmGroupProfile       Get-CdmGroupProfile [-Zone <Cdm...
 ...
```

Managing Delinea Objects Using Windows PowerShell Scripts

This section provides an overview of how to use cmdlets to access and manage authentication and privilege elevation information stored in Active Directory using Windows PowerShell scripts. It provides a summary of the operations you can perform using cmdlets and how to establish a connection to Active Directory. For more examples of how to perform common administrative tasks using the cmdlets, see the samples included with the software.

Using cmdlets to Manage Access

The Delinea access module for PowerShell provides cmdlets that perform operations on objects that correspond to the core elements of Delinea data. Those core elements are:

- Computer role definitions
- Computers
- Groups and group profiles
- NIS network maps and map entries
- Role assignments
- UNIX and Windows rights
- User role definitions
- Users and user profiles
- Zones and zone properties

In most cases, cmdlets can manipulate Delinea information in any type of zone. However, because authorization differs greatly between hierarchical and classic zones, the cmdlets that enable you to work with rights, roles, or role assignments are only applicable in hierarchical zones. You should not use the cmdlets for rights, roles, and role assignments in classic zones. Other than this limitation, you can use the cmdlets to create, access, modify, and remove information associated with any of the core elements of Delinea data for access control and privilege management.

Most of the cmdlets perform one of the following basic operations:

- Add-CdmXxx cmdlets add a right to a specified role.
- Get-CdmXxx cmdlets get the properties of a specified object.
- New-CdmXxx cmdlets create new Delinea objects, such as a new zone or a new role definition.
- Remove-CdmXxx cmdlets delete a specified object or remove a right from a specified role.
- Set-CdmXxx cmdlets set or change the properties of a specified object.

In addition to these basic operations, there are cmdlets for exporting and importing rights and roles from one zone to another and for establishing connections with Active Directory.

For descriptions of the use and parameters for each cmdlet, use the get-help command within the PowerShell console. For example, if you want to see a description and syntax summary for the New-CdmZone cmdlet, type the following command in the PowerShell console:

```
get-help New-CdmZone
```

To see detailed information about a cmdlet's parameters and code examples, you can use the -detailed or -full option. For example, type the following command in the PowerShell console:

```
get-help New-CdmZone -detailed
```

Creating and Using a Connection

Because the Delinea access module for PowerShell cmdlets manipulate objects in Active Directory, you must establish a connection with Active Directory before using cmdlets to perform other tasks. To do that, you must specify a target domain or domain controller and the credentials to use when connecting to that domain or domain controller.

Once the credentials are set, all subsequent calls share that information—you do not have to provide the credential or the domain controller for any subsequent calls.

The following example illustrates how to use the administrator account to connect to the finance.acme domain, then add the user joe.doe to the Engineering zone:

```
PS C:\> Set-CdmCredential "finance.acme" "administrator"
PS C:\> Get-CdmCredential
Target          Type   User
------          ----   ----
finance.acme    Forest administrator@finance.acme
PS C:\> $zone = Get-CdmZone -Name "Engineering"
PS C:\> New-CdmUserProfile -Zone $zone -User "joe.doe@finance.acme" -Login "jdoe"
```

In this example, the cmdlets that get the zone and create the user profile use the credential that is cached by the Set-CdmCredential command. The Get-CdmCredential cmdlet shows what credentials are currently cached.

Managing Connections

You can use the following cmdlets to manage connections to Active Directory by adding, modifying, or using cached credentials or specifying domain-controller-to-domain mappings:

- Set-CdmCredential to add or modify a credential in the cache.
- Get-CdmCredential to list the credentials currently cached.
- Set-CdmPreferredServer to specify a domain controller to use for a domain.
- Get-CdmPreferredServer to list all previously defined domain mappings.

Specifying Credentials

You can use the Set-CdmCredential cmdlet to specify a credential that you want to cache as a PSCredential object. Create the PSCredential object using the Get-Credential cmdlet. The Get-Credential cmdlet prompts users to specify a username and password. You can also pass the username as a parameter to the Get-Credential cmdlet to have the cmdlet prompt the user for the password.

Organizing cmdlet Operations in a Sequence

There is no fixed sequence for calling cmdlets. There is, however, a logical sequence to follow to pass data from one cmdlet to another. For example, to get all of the user UNIX profiles in a zone, you must first identify the zone object before you call the Get-CdmUserProfile cmdlet, To accomplish this, you could organize the calls in the following sequence:

```
$zone = Get-CdmZone -Name "myZone"
Get-CdmUserProfile -Zone $zone
```

Similarly, to get all of the UNIX user profiles for a computer, you must first identify the computer object:

```
$computer = Get-CdmManagedComputer -Name "myComputer"
Get-CdmUserProfile -Computer $computer
```

In most cases, you can determine from the parameters of a cmdlet whether you need to call another cmdlet first. For example, if you want to add a right to a role, you must have created the role first so it can be specified as a parameter to the Add-CdmXxx cmdlet.

For most Set-CdmXxx or Remove-CdmXxx cmdlets, you must call the corresponding Get-CdmXxx or Add-CdmXxx cmdlet to obtain the object first. For example, to delete role1 from zone1, you might call the cmdlets as follows:

```
Get-CdmRole -Zone "cn=zone1,cn=Zones,dc=acme,dc=com" -Name "role1" | Remove-CdmRole
```

In this example, the Get-CdmRole cmdlet retrieves “role1” from the specified zone and passes it to the Remove-CdmRole cmdlet via a PowerShell pipe.

Confirming Licenses

All of the authentication and privilege elevation cmdlets check for a valid license before performing the requested action. The license check succeeds only if there is at least one evaluation, workstation, or server license that has not expired.

If the license check fails, the cmdlet displays an error and stops running. Otherwise, the result is cached. The next time a cmdlet tries to access the same forest, it uses the cached result rather than performing the license check again.

> **Note:** The cache is only effective in one PowerShell console. If another PowerShell console runs a cmdlet accessing the same forest, the cmdlet in that console must perform a separate license check.

Working with Sample Scripts

Introduction

There are several sample scripts included with the software to demonstrate a few common administrative tasks. You can copy and modify these samples to use them in your environment or study them as examples for writing your own custom scripts. The sample scripts include detailed comments about the operations performed to accomplish the following tasks.

Table: Sample Scripts for Administrative Tasks

backup.ps1	How to create a backup copy of a self-contained Delinea zone. This script creates an XML file that contains all computer, user, and group profiles, authorization information, and child zone information for a parent Delinea zone. You cannot use this script to backup SFU zones or child zones.
CreateZoneAndDelegate.ps1	How to create a new zone and delegate all zone administrative tasks to a specific trustee.
RemoveAllOrphans.ps1	How to find and delete all user, group, and computer profiles that no longer have a corresponding Active Directory account on all managed computers in each zone.
RemoveEmptyCompRoles.ps1	How to find and remove computer roles that have no members. This script is only applicable for hierarchical zones.
RemoveEmptyZones.ps1	How to find and remove zones that have no computers, users, or authorization information. This script only removes a zone if it contains no user or group profiles, joined computers, role assignments, computer roles, or child zones. If any of these objects exist for a zone, the zone is not removed. This script is only applicable for hierarchical zones.
ResetOrphanChildZones.ps1	How to find child zones that no longer have a parent zone and reset them as independent zones.
restore.ps1	How to restore a self-contained Delinea zone from a backup created using the backup.ps1 sample script.

Running a Sample Script

To run a sample script:

1. Open the Delinea access module for PowerShell.
2. Verify you have permission to execute scripts by running Get-ExecutionPolicy. In most cases, the permission to execute scripts is restricted.
3. If necessary, use Set-ExecutionPolicy to allow execution. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   Note: For more about execution policies and the options available, run the get-help command.
4. Verify you are in the directory where the scripts are located.
5. Execute the sample script. For example:

   ```
   .\RemoveAllOrphans
   ```

Modifying the Backup and Restore Scripts for Your Needs

If you want to use the sample backup and restore scripts to backup self-contained Delinea zones, you must modify the content of the scripts before executing them. To run a modified sample backup script:

1. Open the backup.ps1 file in a text editor.
2. Modify the path to the zone you want to back up and the path to the backup file at the start of the sample script. For example:

   ```
   $zoneDn = "CN=Headquarters,CN=Zones,OU=Acme Sales,DC=pistolas,DC=org"
   $xmlPath = "C:\Program Files\Centrify\HQ-test.xml"
   ```
3. Modify the confirmation message at the end of the script to display the path to the backup file. For example:

   ```
   Write-Host "Backup to C:\Program Files\Centrify\HQ-test.xml is done."
   ```
4. Save your changes with a new file name, for example, HQbackup.ps1, to keep the sample backup.ps1 script unchanged.
5. Open the Delinea access module for PowerShell.

Using the Default Windows PowerShell Console

Alternatively, you can use the default Windows PowerShell console. If you choose to use that console, run import-module with the path to the access module for PowerShell libraries before performing the above procedure. For example, if you installed the module in the default location, run the following command to import the Delinea access module for PowerShell:

```
import-module "C:\Program Files\Centrify\PowerShell\Centrify.DirectControl.PowerShell.dll"
```

Creating New Zones with the Sample CreateZoneAndDelegate Script

You can use the CreateZoneAndDelegate.ps1 sample script to automate creating new zones and assigning an Active Directory user or group as the zone administrator. By default, the script delegates all administrative tasks to the user or group you specify. To use the script without modification, simply specify the Active Directory container where you want to create the zone, the zone name, and the user or group designated as the zone administrator.

To create new zone using the sample script:

1. Open the Delinea access module for PowerShell.
2. Verify you are in the directory where the scripts are located.
3. Execute the sample script with the required command line arguments. For example:

   ```
   .\CreateZoneAndDelegate -Container "cn=Zones,ou=Acme Sales,dc=pistolas,dc=org" -ZoneName seattle -trustee  frank.smith@pistolas.org
   ```

4. Open Access Manager.
5. Right click **Zones** and select **Open Zone** to search for and select the new zone.
6. If you want to delegate specific administrative tasks, copy the sample script and modify the Set-CdmDelegation call to specify a list of tasks. For example:

   ```
   Set-CdmDelegation -Zone $zone -Task "AddUsers","AddGroups" -Trustee $trustee;
   Write-Host "$trustee is delegated the rights to add users and groups.";
   ```

Generating Reports from Predefined Scripts

Most of the predefined reports in access manager report center have a corresponding PowerShell script that you can use to generate reports from the PowerShell console. See Using Predefined Scripts to Generate Reports for details.

Writing Custom Scripts

Most cmdlets and scripts return information efficiently without any special handling or any noticeable effect on performance. If you plan to write custom scripts that may return large data sets, you should consider ways to improve performance. For example, if you are writing a script that exports a large number of zones or reports on a large number of users, you might want to use the following recommendations as guidelines:

- When testing the performance of the script, use the standard Measure-Command cmdlet to accurately measure cmdlet and script performance.

  Note: The Measure-Command cmdlet ignores the time it takes to print all of the results returned to the PowerShell console. In many cases, the execution of a script is efficient, but rendering the results in the PowerShell console might make the cmdlet or script performance seem unacceptable.

- Consider how you want to balance memory usage and performance when using the PowerShell pipeline if your cmdlet or script returns large data collections.

  For example, you might use foreach in a script instead of using the pipeline to improve performance. Use syntax similar to this:

  ```
  foreach ($cmd in Get-CdmUserProfile -Zone $z) { action_on_each_cmd }
  ```

  Instead of:

  ```
  Get-CdmUserProfile -Zone $z | action_on_each_cmd
  ```

  However, if you choose not to use the pipeline, all of the returned objects stay in memory and might cause an out-of-memory error. Therefore, you should try to maintain balance between the scripts memory usage and performance.

- Cache the data, if possible, by writing the results to a file.

  For example, to add 1000 users to a zone use syntax similar to this to get a zone once:

```
$zone = Get-CdmZone -Dn "cn=QA,cn=Zones,dc=ajax,dc=org" $profile1 = New-CdmUserProfile -Zone $zone -User user1@ajax.org -Uid 10001 ... $profile1000 = New-CdmUserProfile -Zone $zone -User user1000@ajax.org -Uid 11000
```

Instead of using syntax like this, which gets the zone from its distinguished name (DN) for every user:

```
$profile1 = New-CdmUserProfile -Zone "cn=QA,cn=Zones,dc=ajax,dc=org" -User user1@domain.com -Uid 10001 ... $profile1000 = New-CdmUserProfile -Zone "cn=QA,cn=Zones,dc=ajax,dc=org" -User user1000@domain.com -Uid 11000
```

- Use Export-Csv instead of Out-File if possible. The Export-Csv cmdlet writes results to a file faster than the Out-File cmdlet.
- If you are writing a script that generates a very large data set—for example, reporting information for a global zone—you might want to use the native .NET FileStream function. The FileStream function is the fastest way to write content to a file.

  For example, you might use a code snipper like this:

```
$fs = New-Object IO.FileStream <file>, 'Append','Write','Read'
   $fw = New-Object System.IO.StreamWriter $fs
     $zone = Get-CdmZone -Dn "cn=global,cn=Zones,dc=ajax,dc=org"
    foreach ($cz in $zone) {$fw.WriteLine("{0} {1}", $cz.Name, $cz.Type)}
  $fw.Close()
 $fs.Dispose()
```

Enabling Logging for cmdlets

For performance, logging for cmdlets is disabled by default. To enable logging, you must modify the registry on the computer where you are running the access module for Windows PowerShell.

To enable logging:

1. Run regedit to open the Registry Editor
2. Select the HKEY_CURRENT_USER > Software > Delinea registry key.
3. Right-click, then select New > Key and type CIMS.
4. Select the new CIMS key, right-click, then select New > String Value with the name of LogPath.
5. Specify the path to the log file as the value. For example, set the value to C:\Temp\Log.
6. Select the new CIMS key, right-click, then select New > DWORD (32-bit) Value with the name of TraceLevel.
7. Specify the level of detail to write to the log file as the value. The valid settings are:

   0 to disable logging. 1 to only log error messages. 2 to log errors and warning messages. 3 to log errors, warnings, and informational messages. 4 to log all debugging and tracing messages.

   For example, set the value to 4 to enable detailed logging of all messages.

Viewing a Summary of cmdlet Commands

You can use the get-help command with different options to get summary about the cmdlets available in the Delinea access module for PowerShell or detailed information about the specific cmdlets you want to use. For example, you can use get-help with the -full command-line option to see complete reference information for a specified cmdlet or get-help -example to display only the examples for a specified cmdlet.

To see the current list of cmdlets available open the Delinea access module for PowerShell, run the get-help cdm command. This command displays a summary of the access module for PowerShell cmdlets similar to the following table (rendered as ASCII characters):

Table: Summary of cmdlet Commands Output by the get-help cdm Command

Add-CdmApplicationRight	Adds a Windows application right...
Add-CdmCommandRight	Adds a UNIX command right to a s...

Add-CdmDesktopRight	Adds a Windows desktop right to ...
Add-CdmNetworkAccessRight	Adds a Windows network access ri...
Add-CdmPamRight	Adds a PAM application access ri...
Add-CdmSshRight	Adds an SSH application right to...
Export-CdmData	Exports roles and rights from th...
Get-CdmApplicationRight	Gets an application right from a...
Get-CdmCommandRight	Gets a command right from a zone...
Get-CdmComputerRole	Gets a computer role from a zone.
Get-CdmCredential	Gets user credentials.
Get-CdmDesktopRight	Gets a Windows desktop right fro...
Get-CdmEffectiveGroupProfile	Gets effective group profiles fo...
Get-CdmEffectiveUnixRight	Gets the effective UNIX rights a...
Get-CdmEffectiveUserProfile	Gets effective user profiles for...
Get-CdmEffectiveWindowsRight	Gets the effective Windows right...
Get-CdmGroupProfile	Gets group UNIX profiles.
Get-CdmManagedComputer	Gets zoned or auto-zoned managed...
Get-CdmNetworkAccessRight	Gets a Windows network applicati...
Get-CdmNisMap	Gets NIS maps for the specified ...
Get-CdmNisMapEntry	Gets NIS map entries for the spe...
Get-CdmPamRight	Gets a PAM application access ri...
Get-CdmPreferredServer	Gets domain to server mapping.
Get-CdmRole	Gets roles from a zone.
Get-CdmRoleAssignment	Gets role assignments.
Get-CdmSshRight	Gets an SSH application right fr...
Get-CdmUserProfile	Gets user UNIX profiles.
Get-CdmZone	Gets the zone object.
Import-CdmData	Imports roles and rights into a ...
New-CdmApplicationRight	Creates a new Windows applicatio...

New-CdmCommandRight	Creates a new command right in a...
New-CdmComputerRole	Creates a new computer role in a...
New-CdmDesktopRight	Creates a new Windows desktop ri...
New-CdmGroupProfile	Creates a new UNIX group profile.
New-CdmManagedComputer	Pre-creates a computer or comput...
New-CdmMatchCriteria	Creates a new match criteria for...
New-CdmNetworkAccessRight	Creates a new Windows network ac...
New-CdmNisMap	Creates a new NIS map in a speci...
New-CdmNisMapEntry	Creates a new NIS map entry in a...
New-CdmPamRight	Creates a new PAM application ac...
New-CdmRole	Creates a new role in a zone.
New-CdmRoleAssignment	Creates a new role assignment.
New-CdmUserProfile	Creates a new UNIX user profile.
New-CdmZone	Creates a new zone.
Remove-CdmApplicationRight	Deletes a Windows application ri...
Remove-CdmCommandRight	Deletes a command right or remov...
Remove-CdmComputerRole	Deletes a computer role from a z...
Remove-CdmDesktopRight	Deletes a Windows desktop right ...
Remove-CdmGroupProfile	Deletes a UNIX group profile.
Remove-CdmManagedComputer	Removes a managed computer from ...
Remove-CdmNetworkAccessRight	Deletes a Windows network access...
Remove-CdmNisMap	Deletes a NIS map from a zone.
Remove-CdmNisMapEntry	Deletes a map entry from a NIS map.
Remove-CdmPamRight	Deletes a PAM application access...
Remove-CdmRole	Deletes a role.
Remove-CdmRoleAssignment	Deletes a role assignment from a...
Remove-CdmSshRight	Removes an SSH right from a role.
Remove-CdmUserProfile	Deletes a UNIX user profile.

Remove-CdmZone	Deletes an existing zone.
Set-CdmApplicationRight	Updates an existing Windows appl...
Set-CdmCommandRight	Updates an existing command right.
Set-CdmComputerRole	Updates an existing computer role.
Set-CdmCredential	Adds a user credential.
Set-CdmDelegation	Updates the delegation of admini...
Set-CdmDesktopRight	Updates an existing Windows desk...
Set-CdmGroupProfile	Updates an existing UNIX group p...
Set-CdmNetworkAccessRight	Updates an existing Windows netw...
Set-CdmNisMap	Updates an existing NIS map.
Set-CdmNisMapEntry	Updates an existing NIS map entry.
Set-CdmPamRight	Updates an existing PAM applicat...
Set-CdmPreferredServer	Specifies a preferred server.
Set-CdmRole	Updates an existing role.
Set-CdmRoleAssignment	Updates an existing role assignm...
Set-CdmUserProfile	Updates an existing UNIX user pr...
Set-CdmZone	Updates an existing zone.

Objects and Properties

This section lists the objects defined by the authentication and privilege-elevation PowerShell module and the properties of each object.

This chapter provides an alphabetical listing of the objects and the properties of each object defined in the Access module for PowerShell. Note that not all properties are available as parameters in the PowerShell cmdlets.

CdmAdObject Object

Represents an Active Directory object. The following properties are defined for this object.

Table: CdmAdObject Properties

Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
Guid	Guid	Globally unique identifier (GUID) of the Active Directory object.
Name	string	Name of the Active Directory object.

CdmAdPrincipal Object

Represents an Active Directory account principal. The following properties are defined for this object.

Table: CdmAdPrincipal Properties

Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
Guid	Guid	Globally unique identifier (GUID) of the Active Directory object.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	Security identifier (SID) of the Active Directory principal.

CdmApplicationRight Object

Represents a Windows application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmApplicationRight Properties

Description	string	Description of the application right.
IsRequireMfa	Boolean	Indicates whether the application right requires multi-factor authentication.
MatchCriteria	MatchCriteria[]	Filter criteria defined by an array of MatchCriteria objects that identifies the application associated with the application right.
Name	string	Name of the application right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority of the application right; highest priority prevails.
RequirePassword	Boolean	Indicates whether the application right requires authentication.
RunasSelfGroups	group	The group privileges to add to the user's account when running the application associated with the application right.
RunasUser	user	The user to run the application as.
Zone	zone	Zone where the application right is defined.

CdmCommandRight Object

Represents a UNIX command right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmCommandRight Properties

AddVar	string	Comma separated list of environment variable name-value pairs to add to the final list resulting from KeepVar or DeleteVar property (for instance, "var1=a,var2=b,var3=c").
Authentication	string	The authentication type of the command right: none, user, or runastarget.
DeleteVar	string	Comma separated list of environment variables to remove from default set when command is run.
Description	string	Description of the command right.
Digests	string	Specifies SHA-2 digests so that sudo can verify the binary's checksum (SHA-2) before sudo executes the binary. The supported hash types are sha224, sha256, sha384, and sha512.
DzdoRunAsGroup	string	Comma-separated string of groups allowed to run this command using dzdo (for example, "group1,group2,group3"). The asterisk wild card (*) means any group enabled for the zone can run the command. An empty string ("") means the command cannot run as any group.
DzdoRunAsUser	string	Comma-separated list of users allowed to run this command using dzdo (for example, "user1,user2,user3"). - The asterisk wild card (*) means any user enabled for the zone can run the command. An empty string ("") means the command cannot run as any user.
DzshRunas	string	The user this command will run as under dzsh, '$' means current user.
IsAllowNested	Boolean	True if the command is allowed to start another program or open a new shell.
IsDisablePathTraverse	Boolean	True if the command does not allow navigation up the path hierarchy as an argument.
IsPreserveGroup	Boolean	True to retain the user's group membership while executing a command.
IsRequireMfa	Boolean	Indicates whether the command right requires multi-factor authentication.
KeepVar	string	Comma separated list of environment variables to keep in addition to those in dzdo.env_keep when command is run.
MatchPath	string	The path for matching the command.
Name	string	Name of the command right.
Pattern	string	Command pattern for matching the command.
PatternType	string	The type of pattern—glob or regexp—used to match the command.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority for this command; highest priority prevails.
SELinuxRole	string	Sets the SELinux security context to use the specified role when executing a command using dzdo or dzsh. Applies to command rights on Red Hat Enterprise Linux systems that have SELinux enabled and are joined to a hierarchical zone.
SELinuxType	string	Sets the SELinux security context to use the specified type when executing a command using dzdo or dzsh. Applies to command rights on Red Hat Enterprise Linux systems that have SELinux enabled and are joined to a hierarchical zone.
UMask	string	User file-creation mode mask (umask) value that defines who can execute the command.
Zone	CdmZone	Zone of the command right.

CdmComputer Object

Represents an Active Directory computer object. The following properties are defined for this object.

Table: CdmComputer Properties

Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
DNSHostName	string	DNS host name of the Active Directory computer.
Enabled	Boolean	True if the Active Directory computer is enabled.
Guid	Guid	GUID of the Active Directory object.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal.
UserPrincipalName	string	User principal name of the Active Directory computer.

CdmComputerRole Object

Represents a Delinea computer role. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmComputerRole Properties

CustomAttributes	string	Custom text strings for the computer role.
Description	string	Description of the computer role.
Group	CdmGroup	Computer group associated with this computer role.
Name	string	Name of the computer role.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone that contains the computer role.

CdmDesktopRight Object

Represents a Windows desktop access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmDesktopRight Properties

Description	string	Description of the desktop right.
IsRequireMfa	Boolean	Indicates whether the desktop right requires multi-factor authentication.
Name	string	Name of the desktop right.

PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority of the desktop right; highest priority prevails.
RequirePassword	Boolean	True if the desktop right requires a password.
RunasSelfGroups	CdmGroup[]	Groups whose privileges are added to the user account running the desktop.
RunasUser	CdmUser	User to run the desktop as.
Zone	CdmZone	Zone of the desktop right.

CdmEffectiveUnixRights Object

Represents the UNIX rights assigned to a user that are in effect on a Linux or UNIX computer in a zone. The following properties are defined for this object.

Table: CdmEffectiveUnixRights Properties

AuditLevel	string	Effective auditing level.
CommandRights	CdmEffectiveCommandRight []	The array of effective command rights assigned to the user.
Computer	CdmManagedComputer	The computer in which the roles and rights are effective.
HasRescueRight	Boolean	True if the user has the rescue right.
PamRights	CdmEffectivePamRight[]	The array of effective PAM rights assigned to the user.
Profiles	CdmEffectiveUserProfile[]	Effective UNIX profiles for the Active Directory user in the computer.
Roles	CdmEffectiveRole[]	The array of effective roles assigned to the user.
SshRights	CdmEffectiveSshRight[]	The array of effective SSH rights assigned to the user.
UnixSystemRights	string[]	Effective UNIX system rights.
User	CdmUser	Active Directory user assigned to the role.

CdmEffectiveWindowsRights Object

Represents the Windows rights assigned to a user that are in effect on a Windows computer in a zone. The following properties are defined for this object.

Table: CdmEffectiveWindowsRights Properties

AuditLevel	string	Effective auditing level.
ApplicantionRights	CdmEffectiveApplicationRight	The array of effective application rights assigned to the user.
Computer	CdmManagedComputer	The computer in which the roles and rights are effective.

DesktopRights	CdmEffectiveDesktopRight	The array of effective desktop rights assigned to the user.
HasRescueRight	Boolean	True if the user has the rescue right.
NetworkRights	CdmEffectiveNetworkRigh	The array of effective network access rights assigned to the user.
Roles	CdmEffectiveRole	The array of effective roles assigned to the user.
WindowsSystemRights	string[]	Effective Windows system rights.
User	CdmUser	Active Directory user assigned to the role.

CdmGroup Object

Represents an Active Directory group. The following properties are defined for this object.

Table: CdmGroup Properties

Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
GroupCategory	ADGroupCategory	Category of the Active Directory group.
GroupScope	ADGroupScope	Scope of the Active Directory group.
Guid	Guid	GUID of the Active Directory object.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal.

CdmGroupProfile Object

Represents a UNIX group profile. The following properties are defined for this object.

Table: CdmGroupProfile Properties

Computer	CdmManagedComputer	Computer that contains the profile.
Gid	long	GID of the group profile.
Group	CdmGroup	Active Directory group of the group profile.
IsHierarchical	Boolean	True if the group profile is in a hierarchical zone.
IsMembershipRequired	Boolean	True if users are required to be a member of this group.
IsOrphan	Boolean	True if the group profile is an orphan profile, that is, it has no corresponding Active Directory group.

IsSfu	Boolean	True if the group profile is a SFU profile.
Name	string	Name of the group profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone that contains the profile.

CdmLocalGroupProfile Object

Represents a local UNIX group profile. The following properties are defined for this object.

Table: CdmLocalGroupProfile Properties

CanonicalName	string	Canonical name of the local group profile.
Computer	CdmManagedComputer	Computer where the local group profile is defined.
Domain	string	Domain of the local group profile.
Gid	long	GID of the group profile.
Members	string[]	Members of the local group profile.
Name	string	Name of the group profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
State	enum	State of the local group profile. The valid values are: Enable, Remove. and Inherit The default state is Inherit.
Zone	CdmZone	Zone that contains the profile.

CdmLocalUserProfile Object

Represents a local UNIX user profile. The following properties are defined for this object.

Table: CdmLocalUserProfile Properties

CanonicalName	string	Canonical name of the local user profile.
Computer	CdmManagedComputer	Computer where the local user profile is defined.
Domain	string	Domain of the local user profile.
Gecos	string	GECOS field of the local user profile.
HomeDir	string	Home directory of the user associated with the local profile.
Name	string	Name of the user associated with the local profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.

PrimaryGroupId	long	Primary group ID of the user associated with the local profile.
Shell	string	Default shell of the user associated with the local profile.
State	enum	State of the local user profile. The valid values are: Enable, Remove, and Inherit The default state is Inherit.
Uid	long	Numeric user identifier (UID) of the user associated with the local profile.
Zone	CdmZone	Zone where the local user profile is defined.

CdmLocalWindowsGroup Object

Represents a local Windows group account. The following properties are defined for this object.

Table: CdmLocalWindowsGroup Properties

CanonicalName	string	Canonical name of the local group in Active Directory.
Computer	CdmManagedComputer	Computer where the local group is defined.
Description	string	Description for the local group.
Domain	string	Domain of the local group in Active Directory.
Members	string[]	Members of the local group .
Name	string	Name of the local group .
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
State	LocalWindowsGroupState enum	State of the local group . The valid values are: Enable, Remove, and Inherit The default state is Inherit.
Zone	CdmZone	The zone where the local group is defined.

CdmLocalWindowsUser Object

Represents a local Windows user account. The following properties are defined for this object.

Table: CdmLocalWindowsUser Properties

CanonicalName	string	Canonical name of the local user account in Active Directory.
Computer	CdmManagedComputer	Computer where the local user is defined.
Description	string	Description for the local user.
Domain	string	Domain of the local user account in Active Directory.
FullName	string	Full name of the local user.

Name	string	Name of the local user.
PasswordOptions	LocalWindowsUserPasswordOption enum	Password options of the local user. Possible values are: None, Inherit, UserMustChangePasswordAtNextLogon, UserCannotChangePassword, PasswordNeverExpires. It can be a combination of UserMustChangePasswordAtNextLogon and PasswordNeverExpires, UserCannotChangePassword and PasswordNeveExpires.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
State	LocalWindowsUserState enum	State of the local user. he valid values are: Enable, Remove, and Inherit The default state is Inherit.
Zone	CdmZone	The zone where the local user is defined.

CdmManagedComputer Object

Represents a computer managed by authentication and privilege elevation. The following properties are defined for this object.

Table: CdmManagedComputer Properties

AgentVersion	string	Version number of the Delinea Agent installed on the managed computer.
Computer	CdmComputer	Corresponding Active Directory computer account.
ComputerZonePath	string	Path to the computer zone.
IsComputerZoneOnly	Boolean	True if the managed computer has a computer zone only (that is, the computer is not joined to a zone).
IsExpressMode	Boolean	True if the managed computer is in Express (unlicensed) mode.
IsHierarchical	Boolean	True if the managed computer is joined to a hierarchical zone.
IsOrphan	Boolean	True if the managed computer is an orphan profile, that is, it has no corresponding Active Directory computer object.
IsWindows	Boolean	True if the managed computer is a Windows computer.
IsWorkstationMode	Boolean	True if the managed computer is joined to Auto Zone in Workstation mode.
IsJoinedToZone	Boolean	True if the managed computer is joined to a zone.
LicenseType	string	Type of license being used. This property is Server if the managed computer is a Windows or UNIX server or Workstation if the managed computer is not used as a server.
Name	string	Name of the managed computer.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
ScpPath	string	Path to the service connection point for the managed computer.
Zone	CdmZone	Zone of the managed computer.

CdmMatchCriteria Object

Represents an application right match criteria object defined using the application rights match criteria filters. The following properties are defined for this object.

Table: CdmMatchCriteria Properties

Argument	string	The argument for the application.
CompanyName	string	All or part of the company name associated with the application.
CompanyNameMatchOption	string	Specifies whether the company name string should be an exact match (is) or a partial match (contains).
Description	string	The description for the application criteria.
FileDescription	string	All or part of the file description for the application.
FileDescriptionMatchOption	string	Specifies whether the file description string should be an exact match (is) or a partial match (contains).
FileHash	string	The file hash for an application.
FileName	string	The file name for an application.
FileType	string	The file type for an application.
FileVersion	string	All or part of the file version information for an application.
FileVersionMatchOption	string	Specifies whether the file version string should be an exact match (equal), an earlier or equal version (earlier or equal), or a later or equal version (later or equal).
IsArgumentCaseSensitive	Boolean	True if the argument specified is case sensitive.
IsArgumentExactMatch	Boolean	True if the argument must be matched exactly as specified.
IsRequireAdministrator	Boolean	True if the application requires administrator privileges to execute.
LocalOwner	string	The local owner for the application.
LocalOwnerType	string	The local owner type for the application.
OwnerSid	string	The owner security identifier (SID) for the application.
Path	string	The path to an application.
ProductName	string	All or part of the product name associated with the application.
ProductNameMatchOption	string	Specifies whether the product name string should be an exact match (is) or a partial match (contains).
ProductVersion	string	All or part of the product version information for an application.
ProductVersionMatchOption	string	Specifies whether the product version string should be an exact match (equal), an earlier or equal version (earlier or equal), or a later or equal version (later or equal).
Publisher	string	The publisher for an application.
PublisherMatchOption	string	Specifies whether the publisher string should be an exact match (is), a partial match (contains), start with, or end with the specified string.

SerialNumber	string	The serial number for an application.
SerialNumberMatchOption	string	Specifies whether the serial number string should be an exact match (is), a partial match (contains), start with, or end with the specified string.

CdmNetworkRight Object

Represents a Windows network access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmNetworkRight Properties

Description	string	Description of the network right.
IsRequireMfa	Boolean	Indicates whether the network access right requires multi-factor authentication.
Name	string	Name of the network right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority of the network right; highest priority prevails.
RequirePassword	Boolean	True if the network right requires a password.
RunasSelfGroups	CdmGroup[]	Groups whose privileges are added to the user account accessing the network.
RunasUser	CdmUser	Run-as user of the network right.
Zone	CdmZone	Zone of the network right.

CdmPamRight Object

Represents a PAM application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmPamRight Properties

Application	string	PAM application for this right.
Description	string	Description of the PAM access right.
Name	string	Name of the PAM access right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone of the PAM access right.

CdmRole Object

Represents a authentication and privilege elevation role. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmRole Properties

AllowLocalUser	Boolean	True if the role can be assigned to a local user.
AuditLevel	string	Audit setting for this role.
CustomAttributes	string	Custom text strings for the role.
Description	string	Description of the role.
HasRescueRight	Boolean	True if this role can operate without being audited in case of audit system failure.
Name	string	Name of the role.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
RequireMfa	Boolean	True if the role requires multi-factor authentication.
TimeBox	Hashtable	Active time of the role.
UnixSystemRights	string[]	UNIX system rights granted to the role.
WindowsSystemRights	string[]	Windows system rights granted to the role.
Zone	CdmZone	Containing zone.

CdmRoleAssignment Object

Represents a authentication and privilege elevation role assignment. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmRoleAssignment Properties

AdTrustee	CdmAdPrincipal	The trustee, if it is an Active Directory account.
Computer	CdmManagedComputer	Containing computer.
ComputerRole	CdmComputerRole	Containing computer role.
CustomAttributes	string	Custom text strings for the role assignment.
Description	string	The role assignment description.
EndTime	DateTime	The ending date and time for the role assignment.
IsNeverExpire	Boolean	True if the role assignment never expires.
IsRoleOrphaned	Boolean	True if the role is missing or invalid.
IsStartImmediately	Boolean	True if the role assignment starts immediately.
IsTrusteeOrphaned	Boolean	True if the trustee is missing or invalid.
LocalTrustee	string	The trustee, if it is a local account.

PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Role	CdmRole	Assigned role.
StartTime	DateTime	The starting date and time for the role assignment.
TrusteeType	string	Type of trustee.
Zone	CdmZone	Containing zone.

CdmSshRight Object

Represents an SSH application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Table: CdmSshRight Properties

Application	string	Secure shell application for this right.
Description	string	Description of the SSH right.
Name	string	Name of the SSH right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone of the SSH right.

CdmUser Object

Represents an Active Directory user. The following properties are defined for this object.

Table: CdmUser Properties

Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
Enabled	Boolean	True if the Active Directory user is enabled.
GivenName	string	Given name of the Active Directory user.
Guid	Guid	GUID of the Active Directory object.
IsAllADUser	Boolean	True if the user is an Active Directory domain user account.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal
Surname	string	Surname of the Active Directory user.

UserPrincipalName	string	User principal name of the Active Directory user.

CdmUserProfile Object

Represents a UNIX user profile. The following properties are defined for this object.

Table: CdmUserProfile Properties

Computer	CdmManagedComputer	Containing computer.
Gecos	string	GECOS field of the user profile.
HomeDirectory	string	Home directory of the user associated with the profile.
IsHierarchical	Boolean	True if the user profile is in a hierarchical zone.
IsOrphan	Boolean	True if the user profile is an orphan profile, that is, it has no corresponding Active Directory user.
IsSecondary	Boolean	True if the user profile is a secondary profile.
IsSfu	Boolean	True if the user profile is an SFU profile.
IsUseAutoPrivateGroup	Boolean	True if the user private group is to be used as the primary group.
Name	string	Name of the user associated with the profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
PrimaryGroupId	long	Primary group ID of the user associated with the profile.
Shell	string	Default shell of the user associated with the profile.
Uid	long	UID of the user associated with the profile.
UnixEnabled	Boolean	True if the user profile is enabled for a classic zone. This property is not applicable in hierarchical zones.
User	CdmUser	Active Directory user for whom this is the user profile.
Zone	CdmZone	Containing zone.

CdmZone Object

Represents a Delinea zone. The following properties are defined for this object.

Table: CdmZone Properties

AgentlessPasswordAttribute	string	Attribute in which to store the password hash for agentless client.
AvailableShells	string[]	Array of available shells that can be used as the default shell for zone users.
CanonicalName	string	Canonical name of the zone.

CloudInstance	String	Cloud instance URL to which the zone connects.
DefaultGecos	string	Default GECOS field for zone users.
DefaultGid	long	Default GID value to use for zone groups.
DefaultGroupName	string	Default group name to use for zone groups.
DefaultHomeDirectory	string	Default home directory for zone users.
DefaultPrimaryGroup	string	Default primary group to use for zone users.
DefaultShell	string	Default shell for zone users.
DefaultUid	long	Default UID value to use for zone users.
DefaultUserName	string	Default user name to use for zone users.
DefaultValueZone	CdmZone	Zone to use as the source for default values in a selected zone.
Description	string	Description of the zone.
DistinguishedName	string	Distinguished name of the zone.
Domain	string	Active Directory domain associated with the zone.
IsBlockGroupInheritance	Boolean	True if groups defined in a parent zone are not inherited, and therefore not visible, in a child zone. This property is only applicable for hierarchical zones.
IsHierarchical	Boolean	True if it is a hierarchical zone.
IsOrphanChildZone	Boolean	True if the zone is a child zone with no parent zone (Hierarchical zone only).
IsSfu	Boolean	True if it is a SFU zone.
Name	string	Name of the zone.
NextGid	long	Next GID value available for assignment to a zone group.
NextUid	long	Next UID value available for assignment to a zone user.
NisDomain	string	NIS domain for SFU zone or agentless mode.
Parent	CdmZone	Parent zone (Hierarchical zone only).
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
ReservedGid	long	Reserved GID values that cannot be assigned to a zone group.
ReservedUid	long	Reserved UID values that cannot be assigned to a zone user.
Schema	string	Schema of the zone.
SfuDomain	string	SFU domain of the zone (SFU zone only).

TenantId	String	The TenantId of the zone
TruncateUserName	Boolean	True if user names longer than 8 characters are automatically truncated for the zone.
Type	string	Type of the zone.
Variables	string[]	Array of runtime variables.

Adding Users in a One-Way Trust Environment

This section explains how to add a user in a one-way trust environment using the authentication and privilege-elevation PowerShell module.

Some operations, such as adding a user to a zone, may require more than one credential. For example, if you want to add a user from one forest to a zone in another forest when there is a one-way trust between the forest, you might need to specify credentials for each forest. This section explains how to add a user in a one-way trust environment when using PowerShell cmdlets.

Using One Account Credential

If you want to add the user targetuser, who has a domain user account in forest2.net to the zone1 in forest1.net, where forest1.net trusts forest2.net (a one-way trust), you must use an account that has the following permissions:

- Permission to add a user to zone1 in forest1.net.
- Permission to read accounts in forest2.net.

If you have a single account with the appropriate permissions—for example, superuser in forest2.net—you can add the targetuser from forest2.net to the zone1 in forest1.net as follows:

```
Set-CdmCredential "forest1.net" "forest2\superuser"
New-CdmUserProfile -Zone "cn=zone1,cn=Zones,dc=forest1,dc=net" -User "cn=targetuser,cn=Users,dc=forest2,dc=net" -login "UNIXname" -uid nnnnn
```

where UNIXname is the UNIX login name of targetuser and nnnn is the UID of the targetuser.

Using Two Account Credentials

If you do not have a single account with the appropriate permissions in the two forests, adding the targetuser to a zone in another forest will require two account credentials. For example, you must identify accounts with the following permissions:

- An account in forest1.netthat has permission to add a user to zone1 (user1).
- An account in forest2.net that has read permission on forest2.net (user2).

After you identify the accounts with the appropriate permissions—for example, user1 in forest1.net and user2 in forest2.net—you can add the targetuser from forest2.net to the zone1 in forest1.net as follows:

```
Set-CdmCredential "forest1.net" "forest1\user1"
Set-CdmCredential "forest2.net" "forest2\user2"
New-CdmUserProfile -Zone "cn=zone1,cn=Zones,dc=forest1,dc=net" -User "targetUser@forest2.net" -login "UNIXname" -uid nnnnn
```

where UNIXname is the UNIX login name of targetuser and nnnn is the user's UID.

Using Predefined Scripts to Generate Reports

This section describes the predefined report scripts that are included with the authentication and privilege-elevation PowerShell module and how to configure report output files to generate HTML- and PDF-formatted report files.

Most of the predefined reports in Access Manager Report Center have a corresponding PowerShell script to generate reports from the PowerShell console. When you use a PowerShell script to generate a report, the report content displays as text in the PowerShell console window. You can optionally format the report content as an HTML or PDF file using third-party tools.

Provided Report Scripts

The following report scripts are included with authentication and privilege elevation PowerShell. The scripts are typically installed in the following folder:

C:\Program Files\Centrify\PowerShell\Centrify.DirectControl.PowerShell\Reports

For details about script syntax, parameters, and examples, see the script help files. Execute the PowerShell Get-Help command to display the help for a script. For example, to display help details for the ZonesReport.ps1 script, execute the following command from the PowerShell command line:

PS> Get-Help .\ZonesReport.ps1 -Detailed

AuthorizationReportForComputers.ps1	Lists each computer in the zone and indicates which users are allowed to access each computer. This report applies to classic zones only. This report includes details from the user's UNIX profile for each user listed, including the user's Active Directory user name, UNIX user name, zone, UID, shell, home directory, and primary group.	Classic Zone – Authorization Report for Computers
AuthorizationReportForUsers.ps1	Lists each user account in the zone and indicates which computers each user can access. This report applies to classic zones only. This report includes details from the user's UNIX profile for each user listed, including the user's UNIX user name, zone, UID, shell, home directory, and primary group.	Classic Zone – Authorization Report for Users
ComputerEffectiveAuditLevelReport.ps1	Lists the audit level in effect for all authorized users on computers in each zone. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Audit Level
ComputerEffectiveRightsReport.ps1	Lists the privileges granted on each computer. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Rights
ComputerEffectiveRolesReport.ps1	Lists the roles assigned on each computer. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Roles
ComputerRoleAssignmentsReport.ps1	Lists the computer roles that are defined for each zone. The report includes the users and groups and their associated roles. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Role Assignments
ComputerRoleMembershipReport.ps1	Lists the computer roles that are defined for each computer and the zone to which they belong. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Role Membership Report
ComputersReport.ps1	Lists computer account information for each computer in each zone. The information displayed includes the computer account name in Active Directory, the computer's DNS name, the computer's operating system, and the version of the Delinea Agent for *NIX installed on the computer, if available.	Computers Report

GroupsReport.ps1	Lists group information for each group in each zone. The information that is displayed includes the Active Directory group name, the UNIX group name, the UNIX group identifier (GID), and whether the group is an orphan.	Groups Report
StaleComputersReport.ps1	Lists information about all authentication service-enabled computers that have not changed their password in a specified number of days (90 days by default).	Stale Computers Report
UnixUserEffectiveRightsReport.ps1	Lists the effective rights for each UNIX user on each computer. The report shows the name of the right, its type, and where it is defined. This report applies to hierarchical zones only.	Hierarchical Zone – UNIX User Effective Rights
UserAccountReport.ps1	Lists Active Directory account details for the users that have UNIX profiles in each zone. The report includes the Active Directory display name, logon name, and domain for the account. It also includes the account status, such as the date and time of the account's last logon and whether the account is configured to expire, locked out, or disabled.	User Account Report
UsersReport.ps1	Lists information from the UNIX profile for each user in each zone. The report includes the user's Active Directory user name, UNIX user name, UID, shell, home directory, and primary group.	Users Report
WindowsUserEffectiveRightsReport.ps1	Lists the effective rights for each Windows user on each computer. The report shows the name of the right, its type, and where it is defined. This report applies to hierarchical zones only.	Hierarchical Zone – Windows User Effective Rights
ZoneDelegationReport.ps1	Lists the administrative tasks for each zone and the users or groups (trustees) that have been delegated to perform each task. When you grant administrative rights to designated users and groups, you make them "trustees" with permission to perform specific operations. This report indicates which users or groups have permission to perform specific tasks, such as add groups, join computers to a zone, or change zone properties.	Zone Delegation Report
ZoneRolePrivilegesReport.ps1	Lists the roles that are defined for each hierarchical zone and the rights granted by each of these roles, including where each right is defined.	Hierarchical Zone – Zone Role Privileges Report
ZonesReport.ps1	Lists the zone UNIX properties for each zone. This report includes the zone name, list of available shells, the default shell, the default home directory path, the default primary group, the next available UID, reserved UIDs, the next available GID, and reserved GIDs.	Zones Report

Running Report Scripts

When you perform the steps described in this section, the report content displays as text in the PowerShell console window. To generate formatted reports, see Formatting Reports.

To run a report script:

1. Open the Delinea access module for PowerShell reports.

2. Verify you have permission to execute scripts by running Get-ExecutionPolicy. In most cases, the permission to execute scripts is restricted. You can use the Set-ExecutionPolicy to allow execution. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   Note: For more information about execution policies and the options available, use the get-help function.

3. Verify that you are in the directory where the report scripts are located. For example:

   ```
   C:\Program Files\Centrify\PowerShell\Centrify.DirectControl.PowerShell\Reports
   ```

4. Execute the report script. For example:

   ```
   .\ZonesReport.ps1
   ```

Formatting Reports

You can use the following cmdlets to format report output so it can be displayed or processed by third-party tools:

- Export-Csv
- Out-GridView
- Format-Table
- ConvertTo-Html

The following sections describe these cmdlets in detail.

Export-Csv cmdlet

Use this cmdlet to format report output as a CSV file. For example, execute the following command to format the output from the UsersReport.ps1 script as a CSV file:

```
PS> ./UsersReport.ps1 | Export-Csv C:\Report\UsersReport.csv -NoTypeInformation
```

In this example, the output file C:\Report\UsersReport.csv is created, and no type information for the input object is provided. After the CSV file is created, you can open it with third-party applications such as Microsoft Excel.

Out-GridView cmdlet

Use this cmdlet to format report output as an interactive table in a grid view window. For example, execute the following command to format the output from the UsersReport.ps1 script:

```
PS> ./UsersReport.ps1 | Out-GridView
```

Format-Table cmdlet

Use this cmdlet to format report output as a table that is displayed in the PowerShell console window with the selected properties of the object in each column. The object type determines the default layout and properties that are displayed in each column, but you can use the property parameter to select the properties that you want to display. You can specify any of the following parameters on the command line:

- AD User
- Home Directory
- Is Enabled
- Is Orphan
- Primary Group
- Shell
- UID
- UNIX User Name
- Zone

For example, the following command displays the output of UsersReport.ps1 in a table. The -GroupBy option shown here specifies that separate tables are displayed for each zone. Each zone table contains columns for AD User, UNIX User Name, UID, Shell, Home Directory, Is Enabled, Primary Group, and Is

Orphan.

```
. PS> ./UsersReport.ps1 | Format-Table "AD User", "UNIX User Name", "UID", "Shell", "Home Directory", "Is Enabled", "Primary Group", "Is Orphan" -GroupBy Zone
```

Depending on your site's zone configuration, this command would result in output similar to the following:

> **Note:** If the results are too wide to display in the PowerShell console default window size, you can change the PowerShell screen size, and enable some arguments (such as wrap or autosize) provided by this cmdlet.

ConvertTo-Html cmdlet

Use this cmdlet to format report output as an HTML file. This cmdlet returns the result to the PowerShell console window. You can then redirect the result to an HTML file by using the cmdlet Out-File, so that you can read the output using a Web browser. The HTML file created by this cmdlet uses the style sheet defined in the report.css file that is included with authentication and privilege elevation PowerShell.

For example, the following command converts the results of the UsersReport.ps1 script into HTML using the style defined in report.css, and writes the resulting HTML to the output file C:\Report\UsersReport.html.

```
PS> .\UsersReport.ps1 | ConvertTo-Html -CssUri report.css | Out-File C:\Report\UsersReport.html
```

Generating a PDF report

Overview

This section describes how to use the PDFCreator third-party tool to generate PDF output from a report script. The general steps are as follows:

1. Install the PDFCreator third-party tool.
2. Generate HTML output from a report script using the ConvertTo-Html cmdlet.
3. Configure the PDFCreator printer that will convert the HTML output file into a PDF file.
4. Direct the HTML output file to the PDFCreator printer to generate the PDF file.

Procedure Details

The following steps describe how to generate PDF output from the ZonesReport.ps1 script.

1. Note the following
   - You must have administrator privileges to perform these steps.
   - Unless otherwise noted, you perform the steps described here in the PowerShell console window.
   - In this example, the PDF printer that converts HTML to PDF is named "PDFCreator." If the printer has a different name in your environment, use your printer's name.
2. Install PDFCreator from pdfforge.
3. Generate HTML output from the ZonesReport.ps1 script by executing the following command in the PowerShell console:

   ```
   .\ZonesReport.ps1 | ConvertTo-Html -Head "<Style>$(Get-Content .\Report.css)</Style>" | Out-File c:\Reports\ZonesReport.html
   ```

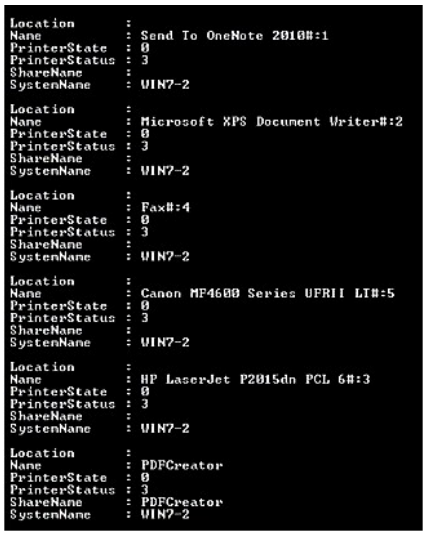
   When you execute this command, the file c:\Reports\ZonesReport.html is created using the styles in Report.css.
4. Specify PDFCreator as the default printer:
   1. Execute the following command to get all installed printers:

      ```
      $printers = gwmi win32_printer
      ```

   2. Run the following variable to list the printers:

      ```
      $printers
      ```

   3. In the list of printers, note the position of the PDFCreator printer in the list. For example, in the following list of printers, PDFCreator is the sixth printer listed:



4. Make PDFCreator the default printer. In this example, because PDFCreator is the sixth printer on the list, you would execute the following command:

   ```
   $printers[5].SetDefaultPrinter()
   ```

5. Ensure PDFCreator is the default printer by clicking **Devices and Printers** on the Windows Start Menu. If PDFCreator is not the default printer, you can make it the default printer there.

5. Configure the auto-save printer settings as follows:

   1. Change the auto-save directory to C:\Reports.
   2. Change the auto-save file name to ZonesReport.
   3. Enable the auto-save feature so that there will be no dialog prompts asking for which file name to save.

6. Perform the following steps to configure the registry to implement these changes. These steps assume that the default registry path is HKCU:\Software\PDFCreator\Program. If your registry path is different, change these commands as appropriate for your environment.

   1. Execute the following command to change the auto-save directory to C:\Reports:

      ```
      Set-ItemProperty -Path "HKCU:\Software\PDFCreator\Program"  -Name "AutoSaveDirectory" -Value "C:\Reports"
      ```

   2. Execute the following command to change the auto-save file name to ZonesReport:

      ```
      Set-ItemProperty -Path "HKCU:\Software\PDFCreator\Program" -Name "AutoSaveFileName" -Value "ZonesReport"
      ```

   3. Execute the following command to enable the auto-save feature:

      ```
      Set-ItemProperty -Path "HKCU:\Software\PDFCreator\Program" -Name "UseAutoSave" -Value "1"
      ```

7. Use Windows Internet Explorer to print the HTML file that you created with the default (PDFCreator) printer. This creates the PDF file.

8. Create and run the following script in the PowerShell console window. The script performs the following tasks:

   1. Creates an IE object and stores it into the $ie variable.
   2. Sets IE output to not display on the screen. This part is optional—if you want IE output to display, you can omit this in the script.
   3. Instructs the $ie object to read the HTML content from the location C:\Reports\ZonesReport.html (the HTML file that you created earlier).
   4. Prints the content of $ie using default printer (PDFCreator), resulting in the generation of the PDF file.

9. The recommended script is as follows:

   ```
   $ie = New-Object -com "InternetExplorer.Application"
   $ie.Visible = $false`
   $ie.Navigate("C:\Reports\ZonesReport.html")
   while ( $ie.busy ) { Start-Sleep -second 1 }
   ```

```
  $ie.ExecWB(6,2)
while ( $ie.busy ) { Start-Sleep -second 1 }
  $ie.quit()
```

Note: This script is specific to the example used in this procedure. If you changed any of the steps in this procedure because of differences in your environment, you might have to make corresponding changes in the script shown.

Auditing and Analysis Scripting Guide

This guide describes the Audit Module for PowerShell command set. These PowerShell cmdlets run on Windows computers and can be used to automate auditing-related management tasks, such as the creation of new audit store databases. You can also use the cmdlets to get or set properties for an installation and perform other administrative tasks. For example, you can write scripts to find and remove sessions matching specific criteria, export audit trail events, or manage audit roles and auditor assignments.

Intended Audience

This guide provides information for auditing infrastructure administrators who want to use PowerShell scripts to manage auditing-related features and components of Server Suite software. This document supplements the help provided within the PowerShell environment using the get-help function. Whereas the get-help function describes each cmdlet in detail, this document provides an introduction to the Auditing Module for Windows PowerShell objects and how you can use PowerShell cmdlets and scripts to perform auditing-related tasks.

This guide assumes general knowledge of PowerShell scripts and syntax, and of the Windows PowerShell modules used to write scripts for Active Directory. You should also be familiar with basic Active Directory operations, such as connecting to a domain controller and managing objects and attributes.

In addition to scripting skills, you should be familiar with Server Suite architecture, terms, and concepts, and know how to perform administrative tasks for the Audit & Monitoring Service and for the platforms you support.

Using this Guide

This guide discusses audit-related administrative tasks using PowerShell-based command-line programs. This information is intended to help you develop scripts for managing the auditing infrastructure, including collectors, audited computers, the audit management database, and the active and attached audit store databases and performing other administrative tasks on Windows computers. With scripts, you can automate the administrative or analytical tasks you might otherwise perform using Audit Manager or Audit Analyzer.

The guide provides the following information:

- Developing Scripts for Administrative Tasks provides an introduction to using Windows PowerShell to perform auditing-related administrative activity.
- Installing the Audit Module for PowerShell describes how to download and install the module as a separate package.
- Managing Audit-Related Objects with Windows Powershell Scripts describes how to use the cmdlets to connect to Active Directory and perform access control and privilege management tasks.
- Auditing-Related Objects and Properties lists the objects defined by the Audit Module for PowerShell, and the properties of each object.

Compatibility and Limitations of This Guide

The information in this guide is intended for use with Server Suite 2015, or later. Although intended to be accurate and up-to-date, interfaces are subject to change without notice and can become incompatible or obsolete when a newer version of the software is released.

In general, application programming interfaces are also intended to be backward-compatible, but are not guaranteed to work with older versions of the software. Because the authentication and privilege elevation cmdlets are subject to change, enhancement, or replacement, the information in this guide can also become incomplete, obsolete, or unsupported in future versions. If you are unsure whether this guide is appropriate for the version of the software you have installed, you can consult the Delinea website or Delinea Support to find out if another version of this guide is available.

Developing Scripts for Administrative Tasks

The Audit Module for PowerShell consists of the following:

- Application programming interfaces in the form of PowerShell command-line programs, or cmdlets, that are packaged in dynamic link libraries (.DLLs).
- A PowerShell help file that includes complete cmdlet reference information and this scripting guide.
- Sample scripts to illustrate administrative tasks.

On Windows computers, you can use the Audit Module for PowerShell to develop your own custom scripts that access, create, or modify auditing components or auditing-related information, such as session activity and audit trail events.

Getting Started with cmdlets For Powershell

The Audit Module for PowerShell consists of "cmdlets" that you can use to manage Server Suite-specific information. A "cmdlet" is a lightweight command-line program that runs in the Windows PowerShell environment. In most cases, cmdlets perform a basic operation and return a Microsoft .NET Framework object to the next command in the pipeline.

The cmdlets in the Audit Module for PowerShell module enable you to access, create, modify, and remove information about Server Suite auditing components and auditing-related information. Using the cmdlets you can manage the entire auditing infrastructure, including installation properties, collectors, audited computers, the audit management database, and the active and attached audit store databases. You can also use cmdlets to manage permissions, audit roles, and role assignments and to work with captured session activity and audit trail events. By combining cmdlets into scripts, you can also automate common administrative tasks, such as the creation of new audit store databases.

Managing Unix Information from a Windows Computer

You can use the cmdlets to work with information for any Server Suite-managed computer where you have enabled the auditing service. However, you can only run the cmdlets on Windows-based computers that have the Windows PowerShell command-line shell available. If you want to develop scripts that run directly on UNIX computers, you can use the ADEdit program (adedit). However, the ADEdit application only provides functionality similar to the cmdlets for access control and privilege management. You cannot use ADEdit to develop scripts for auditing-specific tasks. For detailed information about using ADEdit, see the *ADEdit Command Reference and Scripting Guide*.

Writing Programs in Other Languages

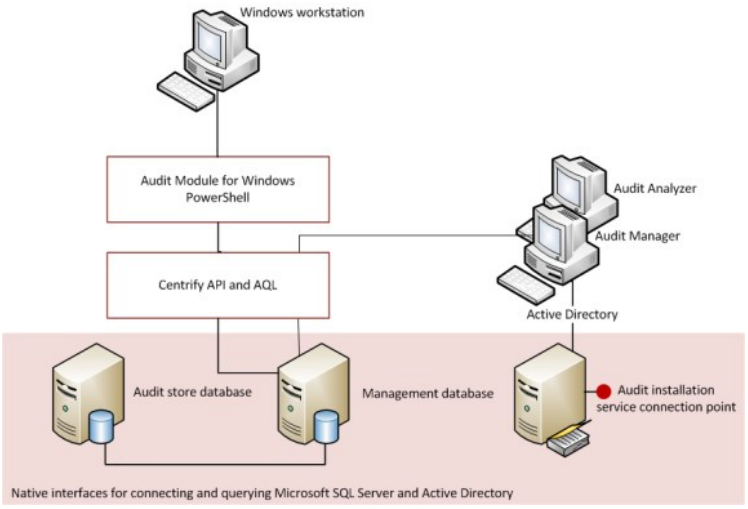
If you want to develop programs or scripts that run on Windows but outside of the Windows PowerShell environment, you can use the Component Object Model (COM) interface that is available as part of the auditing software development kit (SDK). For information about auditing-specific objects you can use the COM-based applications, see the *Database Management Guide*.

Accessing Audit Information Using Native Interfaces

The Audit Module for PowerShell cmdlets connect to Active Directory or to Microsoft SQL Server databases to access audit information. You can, therefore, write PowerShell scripts to automate procedures that you would otherwise perform interactively using Audit Manager or Audit Analyzer.

The cmdlets rely on the underlying interfaces provided by Microsoft Active Directory Service Interfaces (ADSI), Microsoft SQL Server AQL query language, and Server Suite Windows API objects. The ADSI and AQL layers provide low-level functions that permit applications to read and write data. The cmdlets provide a task and object-based level of abstraction for retrieving and manipulating Server Suite audit information so that you do not need to know the details of how the data is stored or how to use any of the underlying ADSI or AQL functions directly.

The following figure illustrates how the Audit Module for PowerShell provides a layer of abstraction between the data stored in Active Directory, the management database, the audit store databases, and your scripting environment.



The Audit Module for PowerShell provides a logical view of the auditing infrastructure and captured information, eliminating the need to know the details of how data is stored in the management database or the audit store databases when performing common administrative tasks. The cmdlets also provide a simple method for accessing audit-related objects without needing to write complex AQL queries.

Using the cmdlets, you can write scripts that automatically create and make active new audit store databases or delete sessions that are no longer of interest. In most cases, the cmdlets enable you to perform exactly the same tasks from the command line that you would otherwise perform interactively using Audit Manager or Audit Analyzer.

Installing the Audit Module for PowerShell

You can install the Audit Module for PowerShell from the Server Suite setup program or as a separate package. It includes the auditing-related cmdlets for Windows PowerShell, sample scripts, and documentation for performing common administrative tasks using PowerShell scripts. This chapter describes how to install the software on a Windows computer.

The following topics are covered:

- About the standalone package
- Running the setup program
- Importing the cmdlets into the Windows PowerShell console

About the Standalone Package

The cmdlets that run in Windows PowerShell are defined in dynamic link libraries that can be installed on any computer where you install other Windows-based components, such as Audit Manager or Audit Analyzer. You can also install these libraries separately, along with sample scripts and documentation, onto computers where no Server Suite software is installed.

If you did not install the Audit Module for PowerShell as a component of a Server Suite installation, you can install it separately. The Audit Module for PowerShell files are available in the same disk image from which you installed Server Suite.

Running the Setup Program

You run the setup program to install the Audit Module for PowerShell files.

To run the standalone setup program

1. Select the downloaded file, right-click, then select **Extract All** to extract the compressed files to a folder.
2. In the Windows disk image for Server Suite, navigate to the /DirectAudit/Powershell folder and double-click the standalone executable to start the setup program.

   For example, double click the Delinea DirectAudit PowerShell64.exe file.
3. At the Welcome page, click **Next**.
4. Select **I accept the terms in the License Agreement**, then click **Next**.
5. Accept the default location or click **Change** to choose a different location, then click **OK**.

   If you accept the default location, the cmdlets are available in a separate Audit Module for PowerShell console.

   If you want the cmdlets to be available in the default Windows PowerShell console with other PowerShell modules, select the following location:

   C:\Windows\System32\WindowsPowerShell\v1.0\Modules\Centrify.DirectAudit.PowerShell
6. Verify the path to the Centrify.DirectAudit.PowerShell folder, then click **Next**.
7. Click **Install**.
8. Click **Finish** to complete the installation.

Importing the cmdlets into the Windows PowerShell Console

If you install the Audit Module for PowerShell in the default location, it is a self-contained Windows PowerShell console. If you install the files in the location for system modules so that cmdlets from other modules are available in the same console, you should import the Audit Module for PowerShell into your default Windows PowerShell console.

> **Note**: According to Microsoft, the Import System Modules task is available only Windows 7 and Windows Server 2008 R2 when Windows PowerShell 3.0 is not installed on the computer. Beginning in Windows PowerShell 3.0, modules are imported automatically the first time that you use a cmdlet in the module.

To import the auditing module

1. On the Start menu, select Windows PowerShell to display a menu extension with a list of Tasks.
2. On the Tasks menu, select **Import System Modules** to import the auditing module and open the Windows PowerShell console.
3. Verify the installation and import completed successfully by typing the following command:

   ```
   get-command *-Cda*
   ```

You should see a listing of the audit cmdlets, similar to the following:

CommandType	Name	Version	Source
Cmdlet	Attach-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Detach-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Export-CdaAuditSessionRecording	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaActiveDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAgent	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditEvent	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditRole	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditRoleAssignment	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditRoleRight	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditSession	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditSessionDataIntegrityStatus	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditSessionReviewer	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditStore	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaAuditStoreRight	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaCollector	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaDetailedExecution	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaInstallation	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaInstallationRight	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaManagementDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaManagementDatabaseRight	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaMonitoredExecution	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaMonitoredFile	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaQuery	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaQueryRight	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaUnixCommand	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaUnixCommandTranscript	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaUserEvent	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Get-CdaWindowsEvent	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	New-CdaAuditRole	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	New-CdaAuditRoleAssignment	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	New-CdaAuditStore	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	New-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	New-CdaSearchCriteria	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Publish-CdaInstallation	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaAgent	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaAuditRole	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaAuditRoleAssignment	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaAuditSession	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaCollector	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Remove-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaActiveDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaAuditRole	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaAuditSession	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaAuditSessionReviewer	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaAuditStore	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaConfiguration	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaInstallation	3.5.2.570	Centrify.DirectAudit.PowerShell
Cmdlet	Set-CdaManagementDatabase	3.5.2.570	Centrify.DirectAudit.PowerShell

Managing Audit-Related Objects with Windows PowerShell Scripts

This chapter provides an overview of how you can use the cmdlets to access and manage audit information stored in Microsoft SQL Server databases and Active Directory using Windows PowerShell scripts. For more examples of how to perform common administrative and auditing analysis tasks using the cmdlets in PowerShell scripts, see the samples included with the software.

Using cmdlets to Manage Auditing

The Audit Module for PowerShell provides cmdlets that perform operations on objects that correspond to the core elements of Server Suite data. The core elements of Server Suite data for auditing are the following:

- Audited computers with the Server Suite auditing services
- Collectors that transfer audited activity from audited computers to the active audit store database
- Active and attached audit store databases
- Management database
- Audit installation
- User sessions
- Audit trail events
- Audit roles
- Audit role assignments

You can use the cmdlets to create, access, modify, and remove information associated with these core elements of Server Suite data for auditing. Most of the cmdlets perform one of the following basic operations:

- New-Cda*Xxx* cmdlets create new Server Suite objects, such as a new audit role or a new audit store database.
- Get-Cda*Xxx* cmdlets get the properties of a specified object.
- Set-Cda*Xxx* cmdlets set or change the properties of a specified object.
- Remove-Cda*Xxx* cmdlets delete a specified object.

In addition to these basic operations, there are cmdlets for attaching or detaching an audit store database, exporting session activity to a file, and for publishing installation information to Active Directory.

For reference information describing the use and parameters for each cmdlet, you can use the get-help function within the PowerShell console. For example, if you want to see a description and syntax summary for the New-CdaAuditStore cmdlet, type the following command in the PowerShell console:

```
get-help New-CdaAuditStore
```

If you want to see more detailed information about a cmdlet's parameters and code examples, you can use the -detailed or -full option. For example, type the following command in the PowerShell console:

```
get-help New-CdaAuditStore -detailed
```

Preparing the Environment to Run cmdlets

Because the Audit Module for PowerShell cmdlets run in the context of a domain account, you don't need to make an explicit connection to an Active Directory domain.

Setting the Preferred Domain Controller

If there is more than one domain controller in the current domain, you can use the SetCdaConfiguration cmdlet to specify the preferred domain controller server to which you want to connect. The following example illustrates how to connect to the preferred domain controller for the finance.acme domain:

```
PS C:\> Set-CdaConfiguration -DomainController "win-2012r2dc.finance.acme"
```

You can also use the SetCdaConfiguration cmdlet to connect to an auditing installation in another trusted forest. In this case, specify the domain controller that is in the other trusted forest.

Setting the Logging Level

You can use the SetCdaConfiguration cmdlet to specify a logging level for running cmdlets. The following example illustrates how to use the Set-CdaConfiguration cmdlet to enable verbose logging:

```
PS C:\> Set-CdaConfiguration -LogLevel "Verbose"
```

The default path to the log file is C:\Program Files\Common Files\Centrify Shared\Logs\DirectAudit_*date-time*.log.

Running cmdlets under Another Account

Some Audit Module for PowerShell cmdlets require permission to connect and update Microsoft SQL Server. If your login credentials do not have the required permissions, you can run cmdlets under another account. To run cmdlets as another user, you can use the standard PowerShell Start-Process -Credential to specify a different user name, then type the user's password when prompted, or you can right-click the Audit Module for PowerShell menu item, then select **Run As Administrator** to run the cmdlets as the local administrator.

Organizing cmdlet Operations in a Sequence

There is no fixed sequence in which cmdlets must be called. There is, however, a logical sequence to follow to make information available from one to another. For example, to get all of the audit roles in an installation, you might first want to identify the installation object you want to work with before you call the Get-CdaAuditRole cmdlet, To accomplish this, you could organize the calls in the following sequence:

```
$site = Get-CdaInstallation -Name "production"
Get-CdaAuditRole $site
```

Similarly, before converting an active database into an attached database, you might organize the calls to create a new audit store database, then set the new database to be the active audit store database:

```
$install = Get-CdaInstallation -Name "site1"

$auditStore = Get-CdaAuditStore -Installation $install -Name "auditstore1"

// Use New-CdaDatabase to create and attach the new database
$newDB = New-CdaDatabase -AuditStore $auditStore -Name "audit-us-Oct2014" -Server "sql_server1.domain.com\da" -Database "audit-us-Oct2014"

// Set the newly created database as the active database for this audit store
Set-CdaActiveDatabase -AuditStore $auditStore -Database $newDB

// Create another new database
$newDB2 = New-CdaDatabase -AuditStore $auditStore -Name "audit-us-Nov2014" -Server "sql_server1.domain.com\da" -Database "audit-us-Nov2014"

// Set the second database as active database
Set-CdaActiveDatabase -AuditStore $auditStore -Database $newDB2

// Detach the first database if it is no longer needed
Detach-CdaDatabase -Database $newDB
```

In most cases, you can determine from the parameters of a cmdlet whether you need to call another cmdlet first. For example, most Set-Cda*Xxx* or Remove-Cda*Xxx* cmdlets, you must call the corresponding GetCda*Xxx* cmdlet to obtain the object first. For example, to delete the "forensics" audit role from the "production" audit installation, you could call the cmdlets as follows:

```
Get-CdaAuditRole -Installation "production" -Name "forensics" | RemoveCdaAuditRole
```

In this example, the Get-CdaAuditRole cmdlet retrieves "forensics" from the specified installation and passes it to the Remove-CdaAuditRole cmdlet.

Checking for Valid Licenses

All of the cmdlets check for a valid license before performing the requested action. The license check succeeds only if one of the following conditions is true:

- There is at least one evaluation license that has not expired.
- There is at least one workstation license.
- There is at least one server license.

If the license check fails, the cmdlet displays an error and stops running. If the license check succeeds, the result is cached. The next time a cmdlet tries to access the same forest, it uses the cached result rather than performing the license check again. Note that the cache is only effective in one PowerShell console. If another PowerShell console runs a cmdlet accessing the same forest, the cmdlet in that console performs a separate license check.

Specifying Parameters using Different Formats

For certain types of parameters, you can specify a value using any one of several different supported formats. For example, you can specify a user principal for a CdaAdPrincipal object type by providing the information that identifies the user in any of the following formats:

- distinguished name (DN) for the user.

- security identifier for the user (SID).
- sAMAccountName attribute for the user in either the *sAMAccountName*@*domain* format or *domain**sAMAccountName* format.
- in a stored user object.

The following formats are all valid for specifying an Active Directory user principal:

```
New-CdaRoleAssignment -AuditRole $role -Assignee "cn=ben,cn=Users,dc=acme,dc=com"

New-CdaRoleAssignment -AuditRole $role -Assignee "S-1-5-21-12345678-98765432-500"

New-CdaRoleAssignment -AuditRole $role -Assignee  "ben@acme.com"

New-CdaRoleAssignment -AuditRole $role -Assignee "acme\ben"

New-CdaRoleAssignment -AuditRole $role -Assignee $userObject
```

The following table lists the supported formats for each type of parameter.

CdaInstallation	You can specify an installation name as string, for example, "DefaultInstallation," or using a CdaInstallation object.
CdaAdPrincipal	You can specify Active Directory users, groups, or computers using any of the following formats: - Distinguished name string - SID string - *sAMAccountName@domain* - *domain**sAMAccountName* You can specify Active Directory users, groups, or computers using a CdaAdPrincipal object.
CdaAccessAccount	You can specify a Windows account name or a SQL Server login account name and password, for example. For a Windows user account, all of the same formats listed for a CdaAdPrincipal object are supported. For SQL Server login accounts, the format is "sql:*sql_name*:*sql_password*". The password can be empty.
CdaAuditScope	You can specify the audit scope using the Active Directory site name as a string, for example, "default-first-site" or by specifying a network subnet definition as a string, for example, "192.168.100.0/24".

If a parameter is not listed in the table, you must specify the object instance returned by a previously cmdlet. For example, you can use the Get-CdaAuditStore cmdlet to return an object instance of the audit store then use that object instance for parameters in other cmdlets that require it.

```
# Get the audit store object instance and store it in $cdaAuditStoreObject

$cdaAduitStoreObject = Get-CdaAuditStore -Installation "DefaultInstallation" -Name "Default-First-Site"

# Use the audit store object instance to specify a parameter value

Attach-CdaDatabase -AuditStore $cdaAduitStoreObject -Name "audit-store-db" -Server "win2012\instance1" -Database "audit-store-database"
```

Working with Sample Scripts

The Audit Module for PowerShell includes some sample scripts that you can use to do database rotation; rotating a database involves creating a new audit store database and making the new database the active database for the installation. You can copy and modify the sample scripts to use the code directly in your environment or study the syntax used in the script to serve as an example for writing your own custom scripts.

- db_rotation.ps1

  This PowerShell script demonstrates how to use the following cmdlets to do database rotation:

  - New-CdaDatabase
  - Set-CdaDatabase
  - Detach-CdaDatabase

- db_rotation_sql_script.ps1

  This PowerShell script demonstrates how to do database rotation using the SQL scripts and PowerShell cmdlets.

- SetupDatabase.sql, SetupServer.sql

  You can modify the file path settings in these SQL scripts to create a new audit store database using the New-CdaDatabase cmdlet with the DatabaseScriptFile and ServerScriptFile parameters. This case is useful if you want to create a new audit store database with a customized database file folder path, database log file folder path, or database full-text catalog root path.

  If you want to use these SQL scripts and you also want to customize the StoredProcedureAccount, IsAwsRds, and IntegrityCheckEnabled settings, you also need to customize the settings in the SQL scripts instead of using the StoredProcedureAccount, IsAwsRds, and IntegrityCheckEnabled parameters. These three parameters can't be used along with the DatabaseScriptFile and ServerScriptFile parameters.

To run the sample script:

1. Open the Audit Module for PowerShell.
2. Verify you have permission to execute scripts.

   ```
   Get-ExecutionPolicy
   ```

   In most cases, the permission to execute scripts is restricted. You can use the SetExecutionPolicy to allow execution. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   For more information about execution policies and the options available, use the gethelp function.
3. Verify you are in the directory where the db_rotation.ps1 script is located.
4. Execute the sample script.

Writing Your Own Scripts

You can combine Audit Module for PowerShell cmdlets with native Windows PowerShell cmdlets to perform many common administrative tasks. The following examples illustrate how you can combine the Audit Module for PowerShell cmdlets and native cmdlets to accomplish a simple but specific goal.

- Exporting Specific Session Fields for a Report
- Checking the Status of Agents and Collectors

Exporting Specific Session Fields For A Report

By default, the cmdlet for getting session information might return more information than you want for a simple report. If you want to narrow down the fields returned, you can use a native cmdlet, such as Select-Object, to specify the fields of interest, then another native cmdlet, such as Out-File, to export the results to a text file. For example, to limit the results to a few key fields, you might specify a command similar to this:

```
Get-CdaAuditSession -Installation "installation-name"
| Select-Object -Property User, Machine, StartTime, EndTime,
State, | Out-File c:\samplesession.txt
```

This example pipes the results from the Audit Module for PowerShell GetCdaAuditSession cmdlet to the Select-Object cmdlet, then uses another pipe to send the resulting output to a file.

Checking the status of agents and collectors

You can use Server Suite cmdlets to get status information for agents and collectors and combine those cmdlets with native or custom cmdlets to schedule checking for connectivity to run on a regular basis and to trigger an email notification if the status returned for the agent or collector in any interval is Disconnected. For example, to check for disconnected agents, you might specify a command similar to this:

```
Get-CdaAgent -i "*installation-name*" | Where { $_.Status -eq "Disconnected" }
```

To check for agents that haven't connected to the collector since a specific time, you might specify a command similar to this:

```
Get-CdaAgent -i "*installation-name*" | where { $_.LastUpdateTime -lt ([DateTime]"12:00:00 AM, 12/29/2014") }
```

Similarly, you can use the Get-CdaCollector cmdlet to check for connectivity between a collector and the Microsoft SQL Server database you are using as the active audit store database.

```
Get-CdaCollector -i "*installation_name*" | where { $_.LastUpdateTime –lt "10:00:00 AM, 12/17/2014"}
```

You can include these cmdlets in scripts that run automatically using a task scheduler to check for connectivity issues at the interval you specify, such as once a day or once a week, and to send the results to specified channels, such as an email message or SNMP trap, using a cmdlet such as Send-MailMessage.

Recommendations for Writing Custom Scripts

Most cmdlets and scripts return information efficiently without any special handling or any noticeable effect on performance. If you plan to write custom scripts that could potentially return large data sets, however, you should consider ways to improve performance. For example, if you are writing a script that exports a large number of sessions or reports on activity for a large audit installation, you might want to use the following recommendations as guidelines.

- When testing the performance of the script, use the standard Measure-Command cmdlet to accurately measure cmdlet and script performance.

  The Measure-Command cmdlet ignores the time it takes to print all of the results returned to the PowerShell console. In many cases, the execution of a query or script is efficient, but rendering the results in the PowerShell console might make the cmdlet or script performance seem unacceptable.

- Avoid using the PowerShell pipeline if your cmdlet or script returns large data collections.

  For example, you might use foreach in a script instead of using the pipeline to improve performance.

  Use syntax similar to this:

  ```
  foreach ($cmd in Get-CdaUnixCommand -Session $s) { *action_on_each_cmd* }
  ```

  Instead of:

  ```
  Get-CdaUnixCommand -Session $s | *action_on_each_cmd*
  ```

- Cache the data, if possible, by writing the results to a file.

  For example, use syntax similar to this:

  ```
  $cmds = Get-CdaUnixCommand -Session $s
  Out-File -InputObject $cmds -FilePath file
  ```

  Instead of:

  ```
  Get-CdaUnixCommand -Session $s | Out-File -FilePath file
  ```

- Use Export-Csv instead of Out-File if possible. The Export-Csv cmdlet writes results to a file faster than the Out-File cmdlet.

- If you are writing a script that generates a very large data set—for example, reporting information for a global audit installation—you might want to use the native .NET FileStream function. The FileStream function is the fastest way to write content to a file.

  For example, you might use a code snipper like this:

  ```
  $fs = New-Object IO.FileStream &lt;file&gt;, 'Append','Write','Read'
    $fw = New-Object System.IO.StreamWriter $fs
          $cmds = Get-CdaUnixCommand -Session $s
          foreach ($cmd in $cmds) {$fw.WriteLine("{0} {1} {2}",
                $cmd.Sequence, $cmd.Time, $cmd.Command)}
    $fw.Close()
  ```

$fs.Dispose() ```

Executing Custom Scripts

In most cases, the permission to execute scripts is restricted. You can use the native PowerShell cmdlet Get-ExecutionPolicy to check whether you have permission to execute scripts using your current account credentials and the native Set-ExecutionPolicy cmdlet to specify an execution policy.

To check and update the execution policy for scripts

1. Open the Audit Module for PowerShell.
2. Verify you have permission to execute scripts.

```
Get-ExecutionPolicy
```

3. Run the SetExecutionPolicy cmdlet to change the execution policy. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   For more information about execution policies and the options available, use the gethelp function.

4. Verify you are in the directory where your scripts are located.

5. Execute the sample script.

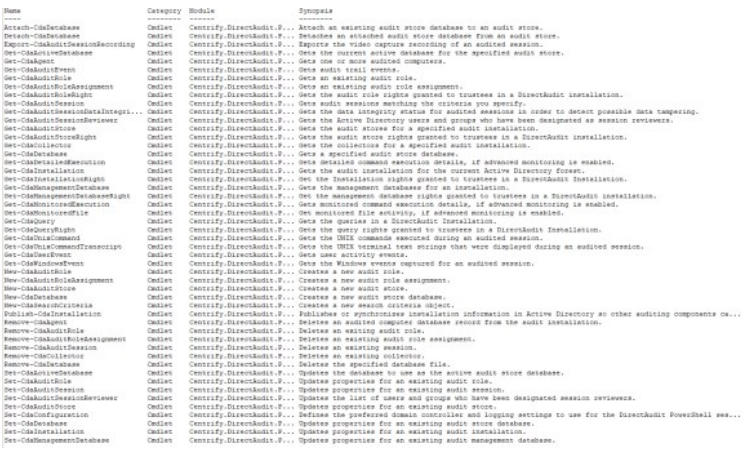
Getting Information about the cmdlet Available

You can use the get-help command with different options to get summary or detailed information about the cmdlets available in the Audit Module for PowerShell or about the specific cmdlets you want to use. For example, you can use get-help with the full command-line option to see complete reference information for a specified cmdlet or get-help -example to display only the examples for a specified cmdlet.

To see the current list of cmdlets available open the Audit Module for PowerShell, then run the following command:

```
get-help *-cda*
```

This command displays a summary of the Audit Module for PowerShell cmdlets similar to the following partial list of commands:



```
Name                                Category  Module                    Synopsis
----                                --------  ------                    --------
Attach-CdaDatabase                  Cmdlet    Centrify.DirectAudit.P... Attach an existing audit store database to an audit store.
Detach-CdaDatabase                  Cmdlet    Centrify.DirectAudit.P... Detaches an attached audit store database from an audit store.
Export-CdaAuditSessionRecording     Cmdlet    Centrify.DirectAudit.P... Exports the video capture recording of an audited session.
Get-CdaActiveDatabase               Cmdlet    Centrify.DirectAudit.P... Gets the current active database for the specified audit store.
Get-CdaAgent                        Cmdlet    Centrify.DirectAudit.P... Gets one or more audited computers.
Get-CdaAuditEvent                   Cmdlet    Centrify.DirectAudit.P... Gets audit trail events.
Get-CdaAuditRole                    Cmdlet    Centrify.DirectAudit.P... Gets an existing audit role.
Get-CdaAuditRoleAssignment          Cmdlet    Centrify.DirectAudit.P... Gets an existing audit role assignment.
Get-CdaAuditRoleRight               Cmdlet    Centrify.DirectAudit.P... Gets the audit role rights granted to trustees in a DirectAudit installation.
Get-CdaAuditSession                 Cmdlet    Centrify.DirectAudit.P... Gets audit sessions matching the criteria you specify.
Get-CdaAuditSessionDataIntegri...   Cmdlet    Centrify.DirectAudit.P... Gets the data integrity status for audited sessions in order to detect possible data tampering.
Get-CdaAuditSessionReviewer         Cmdlet    Centrify.DirectAudit.P... Gets the Active Directory users and groups who have been designated as session reviewers.
Get-CdaAuditStore                   Cmdlet    Centrify.DirectAudit.P... Gets the audit stores for a specified audit installation.
Get-CdaAuditStoreRight              Cmdlet    Centrify.DirectAudit.P... Gets the audit store rights granted to trustees in a DirectAudit installation.
Get-CdaCollector                    Cmdlet    Centrify.DirectAudit.P... Gets the collectors for a specified audit installation.
Get-CdaDatabase                     Cmdlet    Centrify.DirectAudit.P... Gets a specified audit store database.
Get-CdaDetailedExecution            Cmdlet    Centrify.DirectAudit.P... Gets detailed command execution details, if advanced monitoring is enabled.
Get-CdaInstallation                 Cmdlet    Centrify.DirectAudit.P... Gets the audit installation for the current Active Directory forest.
Get-CdaInstallationRight            Cmdlet    Centrify.DirectAudit.P... Get the Installation rights granted to trustees in a DirectAudit Installation.
Get-CdaManagementDatabase           Cmdlet    Centrify.DirectAudit.P... Gets the management databases for an installation.
Get-CdaManagementDatabaseRight      Cmdlet    Centrify.DirectAudit.P... Get the management database rights granted to trustees in a DirectAudit installation.
Get-CdaMonitoredExecution           Cmdlet    Centrify.DirectAudit.P... Gets monitored command execution details, if advanced monitoring is enabled.
Get-CdaMonitoredFile                Cmdlet    Centrify.DirectAudit.P... Get monitored file activity, if advanced monitoring is enabled.
Get-CdaQuery                        Cmdlet    Centrify.DirectAudit.P... Gets the queries in a DirectAudit Installation.
Get-CdaQueryRight                   Cmdlet    Centrify.DirectAudit.P... Gets the query rights granted to trustees in a DirectAudit Installation.
Get-CdaUnixCommand                  Cmdlet    Centrify.DirectAudit.P... Gets the UNIX commands executed during an audited session.
Get-CdaUnixCommandTranscript        Cmdlet    Centrify.DirectAudit.P... Gets the UNIX terminal text strings that were displayed during an audited session.
Get-CdaUserEvent                    Cmdlet    Centrify.DirectAudit.P... Gets user activity events.
Get-CdaWindowsEvent                 Cmdlet    Centrify.DirectAudit.P... Gets the Windows events captured for an audited session.
New-CdaAuditRole                    Cmdlet    Centrify.DirectAudit.P... Creates a new audit role.
New-CdaAuditRoleAssignment          Cmdlet    Centrify.DirectAudit.P... Creates a new audit role assignment.
New-CdaAuditStore                   Cmdlet    Centrify.DirectAudit.P... Creates a new audit store.
New-CdaDatabase                     Cmdlet    Centrify.DirectAudit.P... Creates a new audit store database.
New-CdaSearchCriteria               Cmdlet    Centrify.DirectAudit.P... Creates a new search criteria object.
Publish-CdaInstallation             Cmdlet    Centrify.DirectAudit.P... Publishes or synchronizes installation information in Active Directory so other auditing components ca...
Remove-CdaAgent                     Cmdlet    Centrify.DirectAudit.P... Deletes an audited computer database record from the audit installation.
Remove-CdaAuditRole                 Cmdlet    Centrify.DirectAudit.P... Deletes an existing audit role.
Remove-CdaAuditRoleAssignment       Cmdlet    Centrify.DirectAudit.P... Deletes an existing audit role assignment.
Remove-CdaAuditSession              Cmdlet    Centrify.DirectAudit.P... Deletes an existing session.
Remove-CdaCollector                 Cmdlet    Centrify.DirectAudit.P... Deletes an existing collector.
Remove-CdaDatabase                  Cmdlet    Centrify.DirectAudit.P... Deletes the specified database file.
Set-CdaActiveDatabase               Cmdlet    Centrify.DirectAudit.P... Updates the database to use as the active audit store database.
Set-CdaAuditRole                    Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit role.
Set-CdaAuditSession                 Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit session.
Set-CdaAuditSessionReviewer         Cmdlet    Centrify.DirectAudit.P... Updates the list of users and groups who have been designated session reviewers.
Set-CdaAuditStore                   Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit store.
Set-CdaConfiguration                Cmdlet    Centrify.DirectAudit.P... Defines the preferred domain controller and logging settings to use for the DirectAudit PowerShell ses...
Set-CdaDatabase                     Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit store database.
Set-CdaInstallation                 Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit installation.
Set-CdaManagementDatabase           Cmdlet    Centrify.DirectAudit.P... Updates properties for an existing audit management database.
```

Auditing-Related Objects and Properties

Most Audit Module for PowerShell cmdlets return object instances either directly or as properties of other objects. This section provides an alphabetical listing of the objects and the properties of each object defined in the Audit Module for PowerShell. Note that not all properties are available as parameters in the PowerShell cmdlets.

CdaAccessAccount CdaAdPrincipal CdaAgent CdaAuditEvent CdaAuditRole CdaAuditRoleAssignment CdaAuditRoleRight CdaAuditScope CdaAuditSession CdaAuditSessionTag CdaAuditSessionDataIntegrityStatus CdaAuditStore CdaAuditStoreRight CdaCollector CdaDatabase CdaDetailedExecution CdaInstallation CdaInstallationRight CdaManagementDatabase CdaManagementDatabaseRight CdaMonitoredExecution CdaMonitoredFile CdaQuery CdaQueryRight CdaSearchCriteria CdaUnixCommand CdaUnixCommandTranscript CdaUserEvent CdaWindowsEvent

CdaAccessAccount

Represents a Windows user or SQL Server login account with access to auditing components. The following properties are defined for this object.

AccountName	String	Name of the Windows user or SQL Server login account.
Type	Enum	Account type. The valid values are: 1 if the account is a Windows account that uses Windows authentication. 2 if the account is a Microsoft SQL Server login account that uses SQL Server authentication.

CdaAdPrincipal

Represents an Active Directory principal. The principal can be an Active Directory user, group, or computer account. You can use the Class property to identify the type of principal. Only the account name for the principal is stored in the database. The following properties are defined for this object.

Class	String	Principal type of the Active Directory object.
DistinguishedName	String	Distinguished name of the Active Directory object.
Domain	String	Domain name for the Active Directory principal.
GUID	Guid	Globally unique identifier (GUID) for the Active Directory object.
Name	String	Name of the Active Directory object.
SamAccountName	String	The sAMAccountName attribute for the Active Directory principal.
SID	Security identifier	The security identifier (SID) for the Active Directory principal.

CdaAgent

Represents an audited computer where the auditing service is enabled. The following properties are defined for this object.

AuditedSystemType	Enum	Specifies whether the audited systems are system-based ("SystemBased") or vault-based ("VaultBased"). This parameter is optional. If you do not specify this parameter, the results include a list of both types of audited systems. System-based describes a Windows or UNIX computer that is running an agent. You can access these systems either directly or from the Privileged Access Service Admin Portal. Vault-based describes a Windows or UNIX computer or a network device that is not running an agent (agentless). You can access these systems from the Privileged Access Service Admin Portal. Note: Some properties display different values for vault-based systems: * Version: this property is empty because vault-based systems are agentless * Status: this property displays as none * StartupTime: this property displays as a default date-time value of "1/1/0001 12:00:00 AM" * UpTime: this property displays as 00:00:00

LastUpdateTime	DateTime	Time at which the auditing service agent was last updated.
MachineAddress	String	IP address of the computer hosting the auditing service.
MachineName	String	Name of the computer hosting the auditing service.
MachineSid	String	Security identifier string for the computer hosting the auditing service.
StartupTime	DateTime	Time at which the auditing service agent first started.
Status	Enum	Status of the auditing service. The valid values are: Connected Disconnected
Type	Enum	Type of operating system running on the computer hosting the auditing service. The valid values are: 0 — Unknown 1 — UNIX 2 — Windows
UpTime	TimeSpan	Total time the auditing service agent was connected time from the startup time to the last update time.
Version	String	Auditing service agent version number.

CdaAuditEvent

Represents an audit trail event. The following properties are defined for this object.

Description	String	Description of the audit trail event.
EventId	Integer	Event identifier for the audit trail event. Event instances that share the same event type will also have the same EventId.
EventName	String	Name of the audit trail event.
Machine	String	Computer name associated with the audit trail event.
Parameters	String Array	List of parameters for this audit trail event.
Result	String	Result returned by the audit trail event.
SessionId	String	Identifying string for the session associated with the audit trail event, if there is one.
SessionUri	String	The uniform resource identifier (URI) for the session associated with the audit trail event, if there is one.
Time	DateTime	Date and time the audit trail event occurred.
UniqueId	Long	Unique identifier for the event instance.
User	String	User name associated with the audit trail event.

CdaAuditRole

Represents an audit role. The following properties are defined for this object.

Definition	String	String that defines the criteria used in the audit role to specify the sessions to include.
Description	String	Description of the audit role.
Name	String	Name of the audit role.
Privilege	Enum array	User privileges for the audited sessions that match the criteria specified for this audit role.

CdaAuditRoleAssignment

Represents an audit role assignment. The following properties are defined for this object.

Assignee	CdaAdPrincipal	User, group, or computer account assigned to the audit role.
AuditRole	CdaAuditRole	Name of the audit role being assigned.

CdaAuditRoleRight

This object represents the rights granted to a trustee on the audit role. The following properties are defined for this object.

AuditRole	CdaAuditRole	The audit role
Trustee	String	Trustee name in format <DOMAIN>\<User account name> Note: The consistent name format is shown in the Audit Manager console. For an orphan trustee, it shows SID in SDDL format.
TrusteeType	String	Indicate the type of the trustee,for example Active Directory User or Group
Rights	string[]	The collection of rights granted to the trustee on the audit role. Possible rights: Full Control Change Permissions Change Role Membership Change Role Definition

CdaAuditScope

Represents the audit scope for an audit store. The following properties are defined for this object.

Definition	String	String that defines the audit scope. If the audit scope is an Active Directory site, this property is the site name. If the scope is a subnet, this property is the IP address and subnet mask.
Type	Enum	The type of audit scope. The valid values are: 1 if the audit scope is an Active Directory site. 2 if the audit scope is a network subnet segment.

CdaAuditSession

Represents an audited user session. The following properties are defined for this object.

AuditStore	String	Name of the audit store.

ClientAddress	String	Client IP address.
ClientName	String	Client name.
Comment	String array	Comments that have been added by reviewers to the session.
EndTime	DateTime	Session end time.
IsADUser	Boolean	Indicates whether the user is an Active Directory user.
Machine	String	Host name of the computer where the session ran.
MachineAddress	String	Computer IP address of the computer where the session ran.
MachinePrincipal	String	Computer principal name of the computer where the session ran.
ReviewedBy	ADUser	Name of the user who last updated the review status for the session.
ReviewStatus	Enum	Session review status. The valid values are: 0 for None 1 for ToBeReviewed 2 for Reviewed 3 for PendingForAction 4 for KeepForever 5 for ToBeDeleted
ReviewTime	DateTime	Date and time of the last review status update for the session.
SessionID	String	Globally unique identifier (GUID) for the object.
Size	Integer	Total size of the session in KB.
StartTime	DateTime	Session start time.
State	Enum	Status of the session. The valid values are: -1 for Unknown 0 for InProgress 1 for Terminated 2 for Disconnected 3 for Completed
Tags	String	The tags associated with the audit session
Type	Enum	Session type. The valid values are: 1 if the session is a Windows session 2 if the session is a UNIX session
Uri	String	The uniform resource identifier (URI) for replaying the session in the session player.
User	String	User name associated with the session.
UserDisplayName	String	User display name associated with the session.
Zone	String	Server Suite zone name.

CdaAuditSessionTag

Represents the keyword tag that is associated with an audited session. The following properties are defined for this object.

AuditStoreDatabaseId	int	The ID of the audit store database
Id	long	The ID of the tag

Mode	string	The mode of the tag, which indicates if the session was tagged by an automatic process or manually tagged. The possible values are: Manual Automatic
ReplayTimestamp	DateTime	The session replay time of the tag in the audit session
Session	CdaAuditSession	The audit session(s) that the tag is associated with
Tag	string	The tag
Tagger	string	The user name of the auditor who tagged the audit session
TagTimestamp	DateTime	The timestamp when the audit session was tagged

CdaAuditSessionDataIntegrityStatus

If you've enabled the audit store database for data integrity checking, this object refers to the session's data integrity status. Data integrity checking provides the ability to detect if auditing data has been tampered. The following properties are defined for this object.

Session	CdaAuditSession	The audited user session
Status	Integer	Unknown = -1 Passed = 0 Not Enabled = 1 Session Not Found = 2 Missing Final Thumbprint = 3 Invalid Final Thumbprint = 4 Missing Thumbprint = 5 Invalid Thumbprint = 6 Failed = 7
StatusMessage	String	The friendly display message of the status
Source	String	The source name of the audit session data (which contains the database table name and record Id)

CdaAuditStore

Represents an audit store. The following properties are defined for this object.

Affinity	AffinityType enum	The type agents that the audit store serves, either Windows, UNIX, or both. The possible values are: WindowsAndUnix - 0 Windows - 1 Unix - 2
Name	String	Name of the audit store.
Scopes	CdaAuditScope[]	Audit store scopes.
TrustedAgentEnabled	Boolean	Whether the trusted agent filter is enabled or not.
TrustedAgents	CdaComputer[]	Trusted agent computers.
TrustedCollectorEnabled	Boolean	Whether the trusted collector filter is enabled or not.
TrustedCollectors	CdaComputer[]	Trusted collector service computers.

CdaAuditStoreRight

This object represents the rights granted to a trustee on the audit store. The following properties are defined for this object.

AuditStore	CdaAuditStore	The audit store
Trustee	String	Trustee name in format <DOMAIN>\<User account name> Note: The consistent name format is shown in the Audit Manager console. For an orphan trustee, it shows SID in SDDL format.
TrusteeType	String	Indicate the type of the trustee,for example Active Directory User or Group
Rights	string[]	The collection of rights granted to the trustee on the auditstore. Possible rights: Full Control Change Permissions Modify Name Manage Scopes Manage SQL Logins Manage Collectors Manage Audited Systems Manage Databases Manage Database Trace

CdaCollector

Represents a collector service computer. The following properties are defined for this object.

AuditStoreDatabase	String	The audit store database the collector connects to.
LastUpdateTime	DateTime	The date and time at which the collector received the last update.
MachineAddress	String	IP address of the computer hosting the collector service.
MachineName	String	Name of the computer hosting the collector service.
PortNumber	Integer	Collector connection port number.
Sid	String	Security identifier string for the computer hosting the collector service.
StartupTime	DateTime	The date and time at which the collector first connected to the audit store database.
Status	Enum	Status of the collector service. The valid values are: Connected Disconnected
UpTime	TimeSpan	Total time the collector was connected time from startup to the last update time.
Version	String	Collector service version number.

CdaDatabase

Represents an audit store database. The following properties are defined for this object.

ActiveEndTime	DateTime	The date and time at which the database stopped being the active database.
ActiveStartTime	DateTime	The date and time this database became the active database.
AllowedCollectors	CdaAccessAccount[]	Allowed collector accounts.
AllowedManagementServers	CdaAccessAccount[]	Allowed management database accounts.
AuditStore	CdaAuditStore	The audit store object instance for the database.
CollectorCount	Integer	Number of collectors connected to the database.

Database	String	Microsoft SQL Server database name for the audit store database.
DiskUsage	Integer	Database file size, in 8KB pages.
IsActive	Boolean	Specifies whether this is the active database for the audit store.
Name	String	Display name of the audit store database.
RecordCount	Integer	Number of session records in the database.
Server	String	Microsoft SQL Server host name and instance.
Status	Enum	Database status. The valid values are: Connected Disconnected
Version	String	Database version number.

CdaDetailedExecution

Represents detailed command execution details, if advanced monitoring is enabled. The following properties are defined for this object.

User	String	The user name associated with the event
Machine	String	The computer name associated with the event
Time	DateTime	The date and time when the command was executed
EnteredCommand	String	The name of the entered command
ExecutedCommand	String	The name of the executed command
CommandArguments	String	The command arguments
RunAsUser	String	The run as user name
AccessStatus	String	The access status: Succeeded or Failed
AccessStatusDetails	String	The detailed message about the status
CurrentDirectory	String	The current directory of the command execution
ProcessId	String	The process ID of the command execution
ParentProcessId	String	The process ID of the parent process of the command execution

CdaInstallation

Represents an audit installation. The installation defines the scope of the auditing infrastructure and audit data available for review and play back. The following properties are defined for this object.

		Indicates whether users can delete their own sessions. This installation-wide option takes

DisableSelfDelete	Boolean	precedence over the permissions granted to a user account. If you set this option to be True, users cannot delete their own sessions regardless of the rights granted to their audit roles.
DisableSelfReview	Boolean	Indicates whether users can update the review status or the comments on their own sessions. This installation-wide option takes precedence over the permissions granted to a user account. If you set this option to be True, users cannot update the review status or add comments for their own sessions regardless of the rights granted to their audit roles.
EnableVideoCapture	Boolean	Indicates whether the video capture auditing of user activity is enabled or not.
ManagementDatabase	CdaManagementDatabase	The default connected management database for the installation.
Name	String	Name of the installation.
NotificationImage	String	Name of the notification banner image file in base64 string format.
NotificationMessage	String	Name of the file containing the notification message text.
PublishLocations	String Array	One or more Active Directory locations where the installation service connection point is published.

CdaInstallationRight

This object represents the rights granted to a trustee on the DA Installation. The following properties are defined for this object.

Trustee	String	Trustee name in format <DOMAIN>\<User account name> Note: The consistent name format is shown in the Audit Manager console. For an orphan trustee, it shows SID in SDDL format.
TrusteeType	String	Indicate the type of the trustee,for example Active Directory User or Group
Rights	string[]	The collection of rights granted to the trustee on the DA Installation. Possible rights: Full Control Change Permissions Modify Name Manage Management Database List Manage Audit Store List Manage Collectors Manage Audited Systems Manage Audit Role Manage Queries Manage Publications Manage Licenses Manage Notification Manage Audit Option View

CdaManagementDatabase

Represents an audit management database. The following properties are defined for this object.

AllowedIncomingUsers	CdaUser[]	Allowed incoming users of the management database.
Database	String	Microsoft SQL Server database name for the management database.
Name	String	Display name of the management database.
OutgoingAccount	CdaAccessAccount	Outgoing account of the management database.
Scope	CdaAuditScope[]	Audit store scopes defined for the management database.
Server	String	Microsoft SQL Server host name and instance name.
Status	Enum	Status of the management database. The valid values are: Connected Disconnected

CdaManagementDatabaseRight

This object represents the rights granted to a trustee on the management database. The following properties are defined for this object.

ManagementDatabase	CdaManagementDatabase	The management database
Trustee	String	Trustee name in format <DOMAIN>\<User account name> Note: The consistent name format is shown in the Audit Manager console. For an orphan trustee, it shows SID in SDDL format.
TrusteeType	String	Indicate the type of the trustee,for example Active Directory User or Group
Rights	string[]	The collection of rights granted to the trustee on the management database. Possible rights: Full Control Change Permissions Modify Name Manage Scopes Remove Database Manage SQL Logins Manage Database Trace

CdaMonitoredExecution

Represents monitored command execution details, if advanced monitoring is enabled. The following properties are defined for this object.

User	String	The user name associated with the event
Machine	String	The computer name associated with the event
Time	DateTime	The date and time when the command was executed
Command	String	The name of the executed command
CommandArguments	String	The command arguments
RunAsUser	String	The run as user name
AccessStatus	String	The access status: Succeeded or Failed
AccessStatusDetails	String	The detailed message about the status
CurrentDirectory	String	The current directory of the command execution
ProcessId	String	The process ID of the command execution
ParentProcessId	String	The process ID of the parent process of the command execution

CdaMonitoredFile

Represents monitored file details, if advanced monitoring is enabled. The following properties are defined for this object.

User	String	The user name associated with the event
Machine	String	The computer name associated with the event
Time	DateTime	The date and time when the command was executed

FileName	String	The filename of the file being accessed
Command	String	The name of the executed command
RunAsUser	String	The run as user name
SystemCallName	String	The name of the system call
AccessType	String	The type of the file access: Write or ChangeAttribute
AccessStatus	String	The access status: Succeeded or Failed
AccessStatusDetails	String	The detailed message about the status
CurrentDirectory	String	The current directory of the command execution
ProcessId	String	The process ID of the command execution
ParentProcessId	String	The process ID of the parent process of the command execution

CdaQuery

This object represents a query. The following properties are defined for this object.

Name	String	The query name
Description	String	The query description
IsPredefined	boolean	Whether this query is predefined or not

CdaQueryRight

This object represents the rights granted to a trustee on the query. The following properties are defined for this object.

Query	CdaQuery	The query
Trustee	String	Trustee name in format <DOMAIN>\<User account name> Note: The consistent name format is shown in the Audit Manager console. For an orphan trustee, it shows SID in SDDL format.
TrusteeType	String	Indicate the type of the trustee,for example Active Directory User or Group
Rights	string[]	The collection of rights granted to the trustee on the query. Here are the possible rights: Full Control Change Permissions Read Delete Modify

CdaSearchCriteria

Represents a search criteria object that defines the filters to use to find sessions that can be passed to other cmdlets. For example, you can create a search criteria object to define the sessions that are applicable for a given audit role. The following properties are defined for this object.

Application	String array	Filter sessions by using the Windows application name used.
AuditStore	String	Filter sessions by using the name of the audit store.
ClientName	String	Filter sessions by using the client name of the session.
Comment	String array	Filter sessions by using the comments that have been added by reviewers to the session.
Group	String array	Filter sessions by using the session owner's Active Directory security group.
Installation	String	Filter sessions by using the name of the audit installation.
Machine	String	Filter sessions by using the host name of the computer where the session ran.
ReviewStatus	Enum	Filter sessions by using the session review status. The valid values are: 0 for None 1 for ToBeReviewed 2 for Reviewed 3 for PendingForAction 4 for KeepForever 5 for ToBeDeleted
State	Enum	Filter sessions by using the status of the session. The valid values are: 0 for InProgress 1 for Terminated 2 for Disconnected 3 for Completed
TimeAfter	DateTime	Filter sessions that ran after a specific date and time.
TimeBefore	DateTime	Filter sessions that ran before a specific date and time.
TimeBetween	DateTime	Filter sessions that ran between a start time and an end time.
Type	Enum	Filter sessions by using the session type. The valid values are: 1 if the session is a Windows session 2 if the session is a UNIX session
UnixCommand	String array	Filter sessions by using the UNIX command line input and output.
UnixCommandName	String array	Filter sessions by the UNIX command name only.
UnixCommandTimeAfter	DateTime	Filter sessions that ran after a specific date and time based on the UNIX command input time.
UnixCommandTimeBefore	DateTime	Filter sessions that ran before a specific date and time based on the UNIX command input time.
UnixCommandTimeBetween	DateTime	Filter sessions that ran between a start and end time based on the UNIX command input time.
UnixOutput	Text	Filter sessions by using the UNIX terminal output text captured in the session.
User	String	Filter sessions by using the user name associated with the session.

CdaUnixCommand

Represents an indexed UNIX command captured in an audited session. The following properties are defined for this object.

Command	String	Text of the UNIX command line that was executed.

Sequence	Integer	Sequence number that identifies where in the indexed list of events this event occurs.
Session	CdaAuditSession	The session object.
Time	DateTime	Date and time when the command was executed.

CdaUnixCommandTranscript

Represents the UNIX command input and output captured in an audited session. The following properties are defined for this object.

EndTime	DateTime	The time at which the capture of this command ended.
LineNumber	Integer	The line number at which the text displayed in the terminal.
Role	String	The DirectAuthorize role assigned to this command.
Session	CdaAuditSession	The session object.
StartTime	DateTime	The time at which the capture of this command started.
Text	String	The text displayed in the terminal.
Ticket	String	The trouble ticket assigned to this command.
Type	Enum	Indicates whether the captured text was input or output.

CdaUserEvent

Represents a user event. The following properties are defined for this object.

User	String	User name associated with the user event.
Machine	String	Computer name associated with the audit trail event.
Time	DateTime	The date and time when the command was executed.
Activity	String	A brief description of the user event.

CdaWindowsEvent

Represents an indexed Windows event captured in an audited session. The following properties are defined for this object.

Application	String	Application name associated with the event.
Desktop	String	Desktop name associated with the event if the event occurred when using a desktop access right.
IsAudited	Boolean	Indicates whether this event occurred when using an audited role with a desktop right.

Sequence	Integer	Sequence number that identifies where in the indexed list of events this event occurs.
Time	DateTime	Date and time when the event occurred.
Title	String	Windows title bar text for the application when the event occurred.
Type	Enum	Type of event. The most common event types indicate when a new window or a new application starts or when the title of an existing windows changes.

This chapter explains how to add a user in a one-way trust environment by using the Delinea Windows API.

To add a user in a one-way trust environment, follow these steps:

1. Select an account in a domain that is in a one-way trust relationship with the remote forest so that the account has access to resources in both domains.

   For example, suppose the corporate domain company.corp.com is trusted by the remote domain companyDMZ.com, which is where you intend to add a user. Select an account in the company.corp.com domain that can access resources in the companyDMZ.com domain.

2. Verify that the selected account has permission to modify a zone.

   You can use the zone delegation wizard to add this permission to the selected account. By default, if the user account is a member of the Domain Administrators group in companyDMZ.com, you have the necessary permissions.

3. Use Cims.Connect() to connect to the companyDMZ.com domain to get the Cims object.

4. Obtain an IADsUser object for the remote forest user that you will add to the zone.

   To obtain an IADsUser for company.corp.com using VBScript, for example, use the following code:

   u = GetObject(LDAPCOMPANY.CORP.NETCN=UserName,CN=Users,DC=wonder,DC=land)

   If you log in as a domain user from company.corp.com, you should have sufficient permission.

5. Get the User object by passing the IADsUser object you obtained in the previous step to cims.GetUser(x).

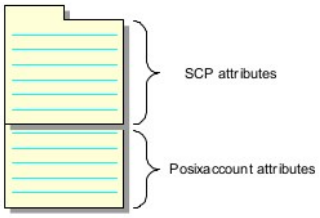
6. With the User object, you can use User.AddUnixProfile() to add the zone profile.

Data Storage for Delinea Zones

This chapter provides a detailed description of how Delinea-specific information is stored in Active Directory. Understanding how this information is stored will enable you to manipulate data for UNIX users and groups using standard LDAP tools, such as ldapadd, directly from UNIX computers, or using Active Directory LDAP utilities, such as adinfo, on Windows computers without using Access Manager, COM objects, or .NET programs.

Classic RFC 2307 Zones (3.x, 4.x)

The classic RFC 2307-compatible zone is similar to the classic Delinea zone, except that the data in the serviceConnectionPoint objects is associated with Active Directory user and group objects stored in RFC 2307-compliant attributes. For RFC 2307-compatible zones, Delinea makes use of a Windows Server feature, called Dynamic Auxiliary Classes, to dynamically bind posixAccount or posixGroup instances to the serviceConnectionPoint objects.



Binding the posixAccount or posixGroup to the user or group serviceConnectionPoint results in an Active Directory object with:

- Two object classes: the serviceConnectionPoint objectClass and the posixAccount or posixGroup objectClass.
- Two sets of attributes: those contributed by the serviceConnectionPoint object and those contributed by posixAccount or posixGroup object.

The structure of the zone and its sub-containers is the same as the classic Delinea zone layout, with each zone stored as a separate tree in the directory and sub-containers for the **Users**, **Groups**, and **Computers** in each zone, but you can use attributes from the posixAccount or posixGroup objectClass to store data in the RFC 2307-compliant format. Storing the data in RFC 2307-compliant attributes enables the information to be used by applications that conform to the RFC 2307 standard.

Zone Attributes in Classic RFC 2307 Zones

The zone object class and its attributes in the classic RFC 2307compliant zone are the same as the classic Delinea zone. The zone object is stored as a container object, and the common name (cn) of the object must be set to the zone name. Most of the other attributes for a zone are stored as pseudo-attributes using the Active Directory description attribute.

The following table summarizes how zone attributes are stored in Active Directory for classic RFC 2307-compliant zones. For more information about any attribute setting, see Zone attributes in classic Delinea zones.

ZoneName	cn:ZoneName For example: cn:default
ZoneVersion	displayName:ZoneVersion The valid values are: ZoneVersion \$CimsZoneVersion3 for RFC 2307 zones, compatible with Delinea, version 3.x \$CimsZoneVersion4 for standard classic zones. \$CimsZoneVersion5 for classic RFC 2307 zones, compatible with Delinea, version 4.x For example: displayName:\$CimsZoneVersion5
Description	description:value For example: description:description:Pilot EMEA
NextUid	description:uidnext:value For example: description:uidnext:12098
NextGid	description:gidnext:value For example: description:gidnext:12098
ReservedUids	description:uidreserved:value For example: description:uidreserved:0-99:501
ReservedGids	description:gidreserved:value For example: description:gidreserved:1000-2500
Availableshells	description:availableshells:value For example: description:availableshells:/bin/sh
DefaultHomeDirectory	description:defaulthome:value For example: description:defaulthome:/nfs/\${user}
DefaultShell	description:defaultshell:value For example: description:defaultshell:/bin/bash
DefaultGroup	description:defaultgid:value For example: description:defaultgid:12098
ZoneType	schema:Dynamic_Schema_Version The valid values for classic RFC 2307compliant zones are: CDC_RFC_2307 (3.x compatible). CDC_RFC_2307_2 (4.x compatible). For example: description: schema:CDC_RFC_2307

User Attributes in Classic RFC 2307 Zones

There are two object classes for the user extension object created in the Users subcontainer of the zone: the serviceConnectionPoint object class and the posixAccount object class.

UnixName	cn:userlogin and uid:userlogin For example: uid:cain
UserVersion	displayName:UserVersion This attribute determines compatibility between a user profile object and the Access Manager console. The only valid value for this attribute is \$CimsUserVersion3. For example: displayName:\$CimsUserVersion3
Uid	uidNumber:value For example: uidNumber:458
Gid	gidNumber:value For example: gidNumber:458
Home	unixHomeDirectory:value For example: unixHomeDirectory:/home/shea
Shell	loginShell:value For example: loginShell:/bin/bash
ParentLink	managedBy:DN_ActiveDirectoryUser If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory user object. For example: managedBy:cn=ben'lau,cn=users,dc=ice,dc=net If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudo-attribute to specify the security identifier (SID) of the parent Active Directory user object. For example: keywords:parentLink:S-n-n-nn-nnn..
UnixEnabled	keywords:unix_enabled:value For example: keywords:unix_enabled:True
ForeignForest	keywords:foreign:value This attribute indicates whether a user in a zone is from an external forest. For example: keywords:foreign:False

Note: The attribute name unixHomeDirectory is not RFC 2307compliant. Microsoft used this name because the attribute homeDirectory was already used in Active Directory.

Group Attributes in Classic RFC 2307 Zones

There are two object classes for the group extension object created in the Groups sub-container of the zone: the serviceConnectionPoint object class and the posixAccount object class.

UnixName	cn:GroupName For example: cn:performx
GroupVersion	displayName:GroupVersion This attribute determines compatibility between a group profile object and the Access manager console. The only valid value for this attribute is \$CimsGroupVersion3. For example: displayName:\$CimsGroupVersion3
Gid	gidNumber:value For example: gidNumber:458
ParentLink	managedBy:DN_ActiveDirectoryGroup If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory group object. For example: managedBy:cn=interns,cn=users,dc=ice,dc=net If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudo-attribute to specify the security identifier (SID) of the parent Active Directory group object. For example: keywords:parentLink:S-n-n-nn-nnn..
UnixEnabled	keywords:unix_enabled:value For example: keywords:unix_enabled:True
ForeignForest	keywords:foreign:value This attribute indicates whether a group in a zone is from an external forest. For example: keywords:foreign:False

Note: The posixGroup group membership attributes are not set. Delinea uses the normal Active Directory mechanism for determining group membership.

Computer Attributes in Classic RFC 2307 Zones

A computer extension object is a serviceConnectionPoint object that is created in the Computers sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

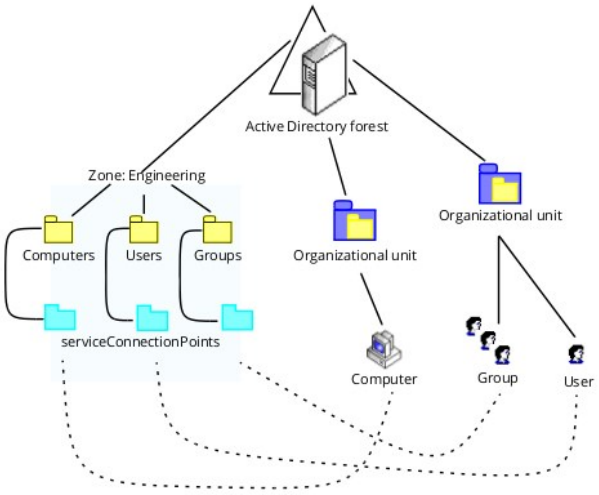
UnixName	name:ComputerName Normally, computer attribute values are set by the adjoin process and do not need to be modified. For example: name:magnolia.ajax.org
ComputerVersion	displayName:ComputerVersion This attribute determines compatibility between a computer profile object and the Access Manager console. The only valid value for this attribute is $CimsComputerVersion3. For example: displayName:$CimsComputerVersion3
ParentLink	managedBy:DN_ActiveDirectoryGroup If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory computer object. For example: managedBy:cn=hr,cn=computers,dc=ajax,dc=org If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudo-attribute to specify the security identifier (SID) of the parent Active Directory computer object. For example: keywords:parentLink:S-n-n-nn-nnn..
AgentVersion	keywords:agentVersion:value For example: keywords:agentVersion:CentrifyDC 4.1 >**Note:** This attribute is set as a keywords attribute if you are using Delinea, versions 3.0.x through 4.1.x. With the Delinea Agent, version 4.2 or later, the AgentVersion is stored using the operatingSystemServicePack attribute of the computer object.
CpuCount	keywords:cpus:value For example: keywords:cpus:2 >**Note:** This attribute is only set if you are using Delinea, versions 3.0.x through 4.1.x. The attribute is not set or updated for computers with the Delinea Agent, version 4.2 or later.
JbossEnabled	keywords:jboss_enabled:value This attribute indicates whether the computer hosts JBoss applications. For example: keywords:jboss_enabled:False
TomcatEnabled	keywords:tomcat_enabled:value This attribute indicates whether the computer hosts Tomcat applications. For example: keywords:tomcat_enabled:False
UnixEnabled	keywords:unix_enabled:value For example: keywords:unix_enabled:True
WebLogicEnabled	keywords:weblogic_enabled:value This attribute indicates whether the computer hosts WebLogic applications. For example: keywords:WEBLOGIC_enabled:True
WebSphereEnabled	keywords:websphere_enabled:value This attribute indicates whether the computer hosts WebSphere applications. For example: keywords:websphere_enabled:True

) [tags]: # (windows api) [priority]: # (1)

Classic Delinea Zones (2.x, 3.x, 4.x)

In classic Delinea zones, each zone is a separate tree stored in the directory. The root of the zone tree is an Active Directory container with the same name as the zone. The zone attributes described in the logical data model are stored in the attributes of this container object. Within the zone container, there are sub-containers for the **Users**, **Groups**, and **Computers** in the zone.

The following figure illustrates the basic structure used for classic zones.



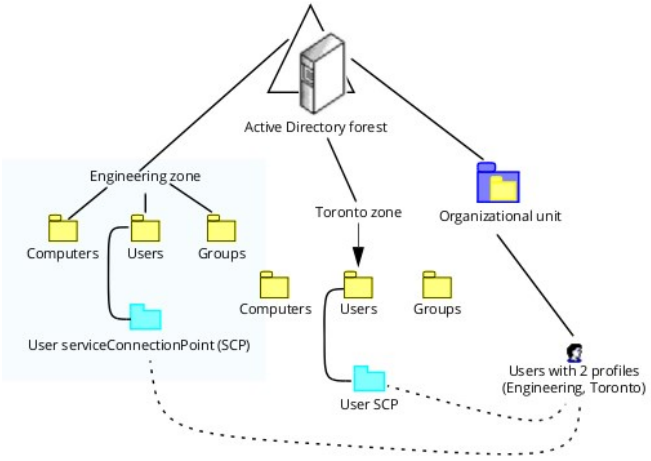
Within each of the sub-containers, there are serviceConnectionPoint (SCP) objects. The serviceConnectionPoint (SCP) objects contain the Delinea attributes for each user, group, or computer defined for the zone. Each of user, group, or computer serviceConnectionPoint objects also has a link back to its parent object (shown as dotted lines in the figure above).

> **Note:** Although Figure 1 illustrates the basic layout for a classic zone using a simple scenario, more complex configurations are possible. For example, in the illustrated scenario, the parent user and group objects are in the same organizational unit (OU), but this is not a requirement. Similarly, the zone tree does not need to be in the same domain as the user or computers objects.

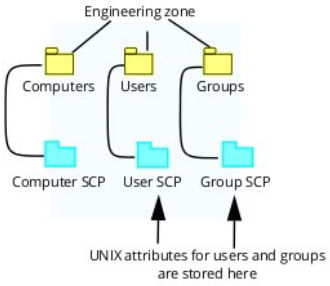
The zone tree structure separates Delinea and UNIX-specific attributes for each zone from every other zone and from the base Active Directory objects for the users and groups. This structure has the following important benefits:

- It enables a single Active Directory user to have many different UNIX profiles.
- It enables you to delegate administrative tasks to users and groups on a zone-by-zone basis.

The following figure illustrates how the zone tree structure enables a single Active Directory user to have many different UNIX profiles.



In a classic zone, the Delinea and UNIX-specific attributes are separate from all of the other zones and from the base Active Directory objects for the users and groups. This enables delegated management of UNIX-related tasks, such as adding or removing UNIX profiles, within each zone.



Parent link Attributes

For each serviceConnectionPoint object in the zone, there is a link back to the corresponding Active Directory object that owns the serviceConnectionPoint object. This link is stored in one of two ways:

- Using the ParentLink pseudo-attribute. This attribute contains the string security identifier (SID) of the parent Active Directory object. This attribute is used in Delinea, version 3.x and later.
- Using the managedBy Active Directory attribute. This attribute is set to the distinguished name (DN) of the parent object. This attribute was used in Delinea, version 2.x, but discontinued because there are cases when it is not possible to set the managedBy attribute.

Zone Attributes in Classic Delinea Zones

The zone object class is stored as a container object. The common name (cn) of the object must be set to the zone name. Most of the other attributes for a zone are stored as pseudoattributes using the Active Directory description attribute. The following table summarizes how zone attributes are stored in Active Directory for Delinea zones.

ZoneName	cn:ZoneName For example: cn:default
ZoneVersion	displayName:ZoneVersion This attribute determines compatibility between a zone object and the Access Manager console. The valid values are: \$CimsZoneVersion2 for zones compatible with Delinea versions 2.x and 3.x \$CimsZoneVersion3 for RFC 2307 classic zones compatible with Delinea versions 2.x and 3.x \$CimsZoneVersion4 for classic zones compatible with Delinea, version 4.x \$CimsZoneVersion5 for hierarchical zones For example: displayName: \$CimsZoneVersion5
Description	description:description:value For example: description:description:Pilot EMEA
NextUid	description:uidnext:value For example: description:uidnext:12098
NextGid	description:gidnext:value For example: description:gidnext:12098
ReservedUids	description:uidreserved:value This attribute can be a multi-valued list, using a colon as the separator. Values can be individual numbers or a range of numbers separated with a dash character (). For example: description:uidreserved:0-99:501
ReservedGids	description:gidreserved:value This attribute has the same format as the reserveduids attribute. For example: description:gidreserved:1000-2500
Availableshells	description:availableshells:value This attribute can be a multi-valued list of shell names, using a colon as the separator. For example: description:availableshells:/bin/sh
DefaultHomeDirectory	description:defaulthome:value For example: description:defaulthome:/nfs/\${user}
DefaultShell	description:defaultshell:value For example: description:defaultshell:/bin/bash
DefaultGroup	description:defaultgid:value For example: description:defaultgid:12098
ZoneType	schema:Dynamic_Schema_Version This attribute identifies the schema layout a zone object uses. The valid values are: Dynamic_Schema_1_0 for Delinea, version 1.0, zones. This schema type is obsolete for version 2.x and later. Dynamic_Schema_2_0 for classic Delinea zones, 2.x and 3.x compatible. Dynamic_Schema_3_0 for classic Delinea zones, 3.x and 4.x compatible. Dynamic_Schema_5_0 for hierarchical Delinea zones, 5.x compatible. SFU_3_0 for SFU zones with the Microsoft Services for UNIX (SFU), version 3.x, schema extension. SFU_4_0 for SFU zones with the Microsoft Services for UNIX (SFU), version 4.x, schema extension. CDC_RFC_2307 for classic RFC 2307compliant zones, Delinea 2.x and 3.x compatible. CDC_RFC_2307_2 for classic RFC 2307compliant zones, Delinea 4.x compatible. CDC_RFC_2307_3 for hierarchical RFC 2307compliant zones, Delinea 5.x compatible. For example: description:schema:Dynamic_Schema_5_0

User Attributes in Classic Delinea Zones

A user extension object is a serviceConnectionPoint object that is created in the Users sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

UnixName	cn:userlogin For SCP objects, the Name attribute is a logical pointer that is the same as the CN attribute. You can use either attribute to store the user's UNIX login name. For example: cn:cain
UserVersion	displayName:UserVersion This attribute determines compatibility between a user profile object and the Access Manager console. The only valid value for this attribute is \$CimsUserVersion2. For example: displayName:\$CimsUserVersion2
ParentLink	managedBy:DN_ActiveDirectoryUser You can use the managedBy or keywords attribute to store the parentLink. If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory user object. For example: managedBy:cn=ben.lau,cn=users,dc=ice,dc=net If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudo-attribute to specify the security identifier (SID) of the parent Active Directory user object. For example: keywords:parentLink:S-n-n-nn-nnn..
Uid	keywords:uid:value For example: keywords:uid:458
Gid	keywords:gid:value For example: keywords:gid:458
Home	keywords:home:value For example: keywords:home:/home/shea
Shell	keywords:shell:value For example: keywords:shell:/bin/bash
UnixEnabled	keywords:unix_enabled:value For example: keywords:unix_enabled:False
ForeignForest	keywords:foreign:value This attribute indicates whether a user in a zone is from an external forest. For example: keywords:foreign:False
AppEnabled	This attribute is no longer used.

Group Attributes in Classic Delinea Zones

A group extension object is a serviceConnectionPoint object that is created in the Groups sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

UnixName	name:GroupName For SCP objects, the Name attribute is the same as the CN attribute. Either attribute can be set, but attribute use should be consistent with other objects. For example: name:performx
GroupVersion	displayName:GroupVersion This attribute determines compatibility between a group profile object and the Access manager console. The only valid value for this attribute is \$CimsGroupVersion3. For example: displayName:\$CimsGroupVersion3
ParentLink	managedBy:DN_ActiveDirectoryGroup If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory group object. For example: managedBy: cn=interns,cn=users,dc=ice,dc=net If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudoattribute to specify the security identifier (SID) of the parent Active Directory group object. For example: keywords:parentLink:S-n-n-nn-nnn..
Gid	gid:value For example: keywords:gid:458
UnixEnabled	This attribute is only applicable in classic 4.x zones.
ForeignForest	Not supported in 3.x or 4.x.

Computer Attributes in Classic Delinea Zones

A computer extension object is a serviceConnectionPoint object that is created in the Computers sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

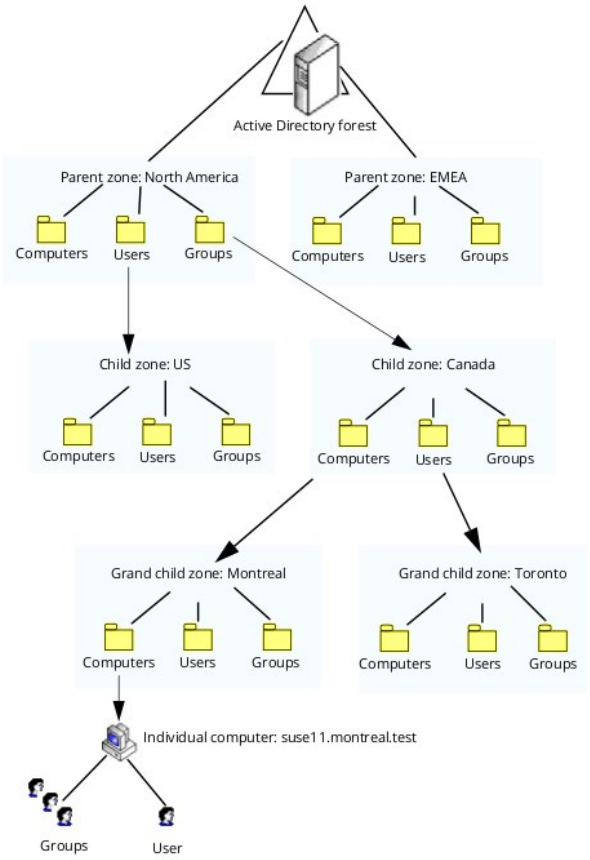
UnixName	name:ComputerName For SCP objects, the Name attribute is the same as the CN attribute. Either attribute can be set, but attribute use should be consistent with other objects. Normally, computer attribute values are set by the adjoin process and do not need to be modified. For example: name:magnolia.ajax.org
ComputerVersion	displayName:ComputerVersion This attribute determines compatibility between a computer profile object and the Access Manager console. The only valid value for this attribute is \$CimsComputerVersion3. For example: displayName:\$CimsComputerVersion3
ParentLink	managedBy:DN_ActiveDirectoryGroup If the zone is a 2.x and 3.x compatible zone, you should set this attribute to the DN of the parent Active Directory computer object. For example: managedBy:cn=hr,cn=computers,dc=ajax,dc=org If the zone does not need to be compatible with older versions of Delinea software, you can use the keywords attribute and parentLink pseudo-attribute to specify the security identifier (SID) of the parent Active Directory computer object. For example: keywords:parentLink:S-n-n-nn-nnn
AgentVersion	keywords:agentVersion:value For example: keywords:agentVersion:CentrifyDC 4.1 **Note** This attribute is set as a keywords attribute if you are using Delinea, versions 3.0.x through 4.1.x. With the Delinea Agent, version 4.2 or later, the agentVersion is stored using the operatingSystemServicePack attribute of the computer object.
CpuCount	keywords:cpus:value For example: keywords:cpus:2 **Note** This attribute is only set if you are using Delinea, versions 3.0.x through 4.1.x. The attribute is not set or updated for computers with the Delinea Agent, version 4.2 or later.
JbossEnabled	keywords:jboss_enabled:value This attribute indicates whether the computer hosts JBoss applications. For example: keywords:jboss_enabled:False
TomcatEnabled	keywords:tomcat_enabled:value This attribute indicates whether the computer hosts Tomcat applications. For example: keywords:tomcat_enabled:False
UnixEnabled	keywords:unix_enabled:value For example: keywords:unix_enabled:True
WebLogicEnabled	keywords:weblogic_enabled:value This attribute indicates whether the computer hosts WebLogic applications. For example: keywords:jboss_enabled:True
WebSphereEnabled	keywords:websphere_enabled:value This attribute indicates whether the computer hosts WebSphere applications. For example: `keywords:websphere_enabled:True

) [tags]: # (windows api) [priority]: # (1)

Hierarchical Delinea Zones (5.x)

In hierarchical Delinea zones, each zone is part of a tree of zones and Active Directory containers for users, groups, and computers. The effective profile for each user, group, or computer in a hierarchical zone is determined by attributes defined in the current zone, in parent zones, and in zone default values, with values lower in the hierarchy taking priority. In addition, individual computers can have computer-level overrides of users, groups, and role assignments. When you assign computer-level overrides for a specific computer, internally Delinea creates a *computer-specific zone* that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager, but are treated as end nodes in the zone hierarchy.

The following figure illustrates the basic structure used for hierarchical Delinea zones. Attributes are inherited from higher-level to lower-level zones as indicated by the arrows in the figure.



In hierarchical zones, because User and Zone objects inherit from their parent zones, it is not necessary for every hierarchical level to have a full set of attributes. If a user profile is incomplete after inheriting all the ancestor's attributes, missing mandatory attributes are filled from defaults. Missing UIDs are generated from an RID-based algorithm.

See Classic Delinea zones (2.x, 3.x, 4.x) for a general description of the way Delinea attributes are stored in Active Directory.

Zone Attributes in Standard Hierarchical Zones

The zone object class is stored as a container object. The common name (cn) of the object must be set to the zone name. Most of the other attributes for a zone are stored as pseudoattributes using the Active Directory description attribute. The following table summarizes how zone attributes are stored in Active Directory for hierarchical Delinea zones.

ZoneName	cn:ZoneName For example: cn:global	No
Description	description:description:value For example: description:description:Pilot-NA	No
AvailableShells	description:availableshells:shell1:shell2 For example: description:availableshells:/bin/sh	Yes
DefaultShell	description:defaultshell:value or description:defaultshell:%{shell} For example: description:defaultshell:/bin/bash	Yes
DefaultHomeDirectory	description:defaulthome:value or description:defaulthome:%{home}/%{user} For example: description:defaulthome:/nfs/jsmith	Yes
UserDefaultGecos	description:defaultgecos:\${u:cn} For example: description:defaulttgecos:\${u:upn}	Yes
customVariable	description:%variablename:value One for each variable. For example: description:%admin:sAMAccountName	Yes
ReservedUids	description:uidreserved:value This attribute can be a multi-valued list, using a colon as the separator. Values can be individual numbers or a range of numbers separated with a dash character (). For example: description:uidreserved:0-99:501	Yes
ReservedGids	description:gidreserved:value This attribute has the same format as the reserveduids attribute. For example: description:gidreserved:1000-2500	Yes
UserDefaultUid	description:defaultuid:value Set value to \${uidnext} to use the zone's cram attribute uidnext. The cram attribute is where the key-value pairs ("name:value") are stored. Set value to \${autosid} to generate the UID from the domain SID and user RID. For example: description:defaultuid:\${autosid}	Yes
DefaultGroup	description:defaultgid:value Set value to -1 to use private groups. For example: description:defaultgid:12098	Yes
UserDefaultName	description:username:\${u:sAMAccountName}	Yes
UserDefaultRole	description:defaultrole:role-name	Yes
GroupDefaultGid	description:defaultgroupgid:value Set value to \${gidnext} to use the zone's cram attribute gidnext in classic zones. Set value to \${autosid} to generate the GID from the domain SID and group RID in hierarchical zones. For example: description:defaultgid:\${autosid}	Yes
GroupDefaultName	description:groupname:\${g:CN}	Yes
NISDomain	description:nisdomain:name	Yes
Schema	description:schema:name Possible values are: CDC_RFC_2307 (for a classic RFC 2307 zone) CDC_GENERIC (for a classic Delinea zone) SFU_3_0 (For a classic SFU-compliant R2 schema zone) SFU_3_0V1 (For a classic SFU-compliant zone) For example: description:Cchema:DC_GENERIC	No
AgentlessAttribute	description:pwsync:attributeName For example: description:pwsync:msSFU30Password	Yes
Licenses	description:license:guid	Yes
SFUDomain	description:alternateDomain:domain.name This is a multi-value attribute. Multi-value attributes are possible because the keyword and value are combined, making each line of the description-keyword string unique.	Yes
Parent	description:parentLink:MS-GUID@DOMAIN.NAME For example: samAccountName@domain.name[:N]: "joe@ajax.com"	No

objectType	displayName=\$CimsZoneVersionnumber where the zone version number can be: \$CimsUserVersion4 for a Delinea zone \$CimsUserVersion5 for a RFC 2307 zone	No

User Attributes in Hierarchical Zones

A user extension object is a serviceConnectionPoint object that is created in the Users sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

cn	sAMAccountName@domain.name[:*N*]	No
objectType	displayName=\$CimsUserVersion4	No
Name	keywords:login:name For example: keywords:login:cain	Yes
Uid	keywords:uid:value For example: keywords:uid:458	Yes
Gid	keywords:gid:value For example: keywords:gid:458	Yes
Home	keywords:home:value For example: keywords:home:/home/shea	Yes
Shell	keywords:shell:value For example: keywords:shell:/bin/bash	Yes
Gecos	gecos:value For example: gecos:%{u:displayName}	Yes

User and group extended attributes are specific to a particular computer and can be set on a per-user or per-group basis. The format for extended attributes depend on the format required for a particular operating system. Currently, only AIX extended attributes are supported.

Each attribute name starts with a prefix that indicates the operating system to which it applies (for example, aix.) and is followed by the attribute name. The valid values for each attribute depend on the attribute type, and can be a string, number or Boolean value. Attributes that support multiple values are specified with separate namevalue pairs.

The specific user and group extended attributes that are available for you to set depend on the version of the operating system running on the computer where the attributes are used. For detailed information about the extended attributes available and valid values on a specific version of the AIX operating system, see your AIX documentation.

The following table lists some of the most commonly-used **user extended attributes** for illustration purposes. It does not represent the complete list of user and group extended attributes that might be available on any given version of the operating system.

aix.admin	Specifies the administrative status of the user as true or false.
aix.admgroups	Lists the groups that the user administrates as a comma-separated list of group names.
aix.daemon	Specifies whether the user can execute programs using the the cron daemon or the system resource controller (src).
aix.rlogin	Specifies whether the user account can be logged into remotely using telnet or rlogin.
aix.su	Indicates whether other users can switch to the user account with the su command.
aix.sugroups	Lists the groups can switch to the user account as a comma-separated list of group names.
aix.tpath	Indicates the user's trusted path status.
aix.ttys	Lists the terminals that can access the account as a comma-separated list of full path names, or using ALL to indicate all terminals.
aix.fsize	Sets the soft limit for the largest file a user's process can create or extend or a value of -1 to specify unlimited for this attribute.

aix.core	Sets the soft limit for the largest core file a user's process can create or a value of -1 to specify unlimited for this attribute.
aix.cpu	Sets the soft limit for the maximum number of seconds of system time that a user's process can use or a value of -1 to specify unlimited for this attribute.
aix.data	Sets the soft limit for the size of a user's data segment or a value of -1 to specify unlimited for this attribute
aix.rss	Sets the soft limit for the largest amount of physical memory a user's process can allocate or a value of 1 to specify unlimited for this attribute.
aix.stack	Sets the soft limit for the largest process stack segment for a user's process or a value of 1 to specify unlimited for this attribute.
aix.nofiles	Sets the soft limit for the number of file descriptors a user process can have open at one time or a value of 1 to specify unlimited for this attribute.
aix.umask	Determines file permissions for the user using a three-digit octal value such as 022.

Group Attributes in Hierarchical Zones

A group extension object is a serviceConnectionPoint object that is created in the Groups sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

version	displayName=\$CimsZoneVersion4 or, for RFC 2307 objects, use version 5: displayName=\$CimsZoneVersion5	No
name	keywords:login:Name For example: keywords:login:ibmdba	Yes
gid	keywords:gid:value For example: keywords:gid:458 If the group is in a standard zone, the GID is stored as gid:xxx in the keywords attribute. If the group is in an RFC 2307 zone, the GID is stored in the schema's gid attribute.	Yes
parentLink	keywords:parentLink:MS-SID For example: keywords:parentLink:S-1-5-21-387451290	No
IsMembershipRequired	keywords:required:value For example: keywords:required:true	Yes
InheritFromParent	keywords:inherit:value For example: keywords:inherit:true	No

Computer Attributes in Hierarchical Zones

A computer extension object is a serviceConnectionPoint object that is created in the Computers sub-container of the zone. The pseudoattributes for this object are stored in the keywords attribute.

unixName	cn:name:ComputerName Normally, the computer attribute values are set by the adjoin process and do not need to be modified.
schemaVersion	displayName:ComputerVersion This attribute determines compatibility between a computer profile object and the Access Manager console. The only valid value for this attribute is \$CimsComputerVersion3. For example: displayName:\$CimsComputerVersion3
agentVersion	keywords:agentVersion:Version For example: keywords:agentVersion:CentrifyDC 5.4.0-197
parentLink	keywords:parentLink:MS-SID For example: keywords:parentLink:S-1-5-21-387451290-2189

Computer-Specific Zone Attributes in Standard Hierarchical Zones

A computer zone object is a zone object that contains computerspecific users and groups. This object is a special type of hierarchical Zone container with computerName:zone and version displayName=\$CimsComputerZoneVersion1.

User Attributes in RFC 2307-Compliant Zones

If you create a hierarchical zone when using the RFC 2307-compliant schema, user attributes are stored in much the same way as in a classic zone with user object attributes stored in corresponding Active Directory attributes.

cn	sAMAccountName@domain.name[:N]	No
Version	displayName=\$CimsUserVersion5	No
Name	uid	Yes
Uid	uidNumber	Yes
Gid	gidNumber	Yes
Home	unixHomeDirectory	Yes
Shell	loginShell	Yes
Gecos	gecos	Yes

For more information, see User attributes in classic RFC 2307 zones.

Group Attributes in RFC 2307-Compliant Zones

If you create a hierarchical zone when using the RFC 2307-compliant schema, group attributes are stored in much the same way as in a classic zone with group object attributes stored in corresponding Active Directory attributes.

Version	displayName=\$CimsUserVersion5	No
Name	keywords=login:Name	Yes
Gid	gidNumber	Yes
Gecos	gecos	Yes

For more information, see Group attributes in classic RFC 2307 zones.

User Attributes in Hierarchical SFU Zones

If you create a hierarchical zone when using the Microsoft Services for UNIX (SFU) schema, user attributes are stored in the same way as in a classic SFU zone with the following exceptions:

- If the SFU schema is version 3.x, the Gecos field is stored in the msSFU30Gecos Active Directory attribute.
- If the SFU schema is version 4.x, the Gecos field is stored in the gecos Active Directory attribute.

For more information, see User attributes in classic SFU-compliant zones.

Group Attributes in Hierarchical SFU Zones

If you create a hierarchical zone when using the Microsoft Services for UNIX (SFU) schema, group attributes are stored in the same way as in a classic SFU zone. For more information, see Group attributes in classic SFU-compliant zones.

The Logical Data Model for Objects

This section describes the logical data model associated with each type of Delinea object. The data types used in the discussion of the logical data model, however, may not reflect the actual implementation of the data for a given zone type. For example, a user's UID value might be stored as an integer in SFU zones or as a string in Delinea zones, but represented as an integer (int) in the logical data model.

Similarly, the names used in the logical data model may not reflect the actual Active Directory attribute names for a given zone type. For example, in Delinea zones, there are UNIX-specific attributes, such as the UID value, that are stored in an Active Directory object where the schema does not have a corresponding attribute, whereas the schema for SFU zones provides this attribute.

Use of Existing Attributes

Delinea uses existing Active Directory attributes to store data. For example, most Delinea zones use Active Directory serviceConnectionPoint objects to store UNIX-specific data. The serviceConnectionPoint class is intended to hold information about services. The keywords attribute of the serviceConnectionPoint object holds name-value pairs that an Active Directory service can use to store its own attributes.

For example, if you were to use ldapsearch to filter the keywords attribute for a user's serviceConnectionPoint class in a Delinea zone, you would see results similar to the following:

```
keywords: foreign:False
keywords: gid:800 keywords: home:/home/jae
keywords: parentLink:S-1-5-21-3619765212-102450798-26543
keywords: shell:/bin/bash
keywords: uid:810
keywords: unixEnabled:True
```

Once you are familiar with the logical data model for Delinea objects, refer to the appropriate zone-specific section for more detailed information about which Active Directory attributes are used to store data in a particular type of zone.

Logical Data Attributes for Zones

The following table describes the logical attributes for Delinea zone objects.

NextUid	int	The value of the next UID that will be allocated automatically.
NextGid	int	The value of the next GID that will be allocated automatically.
ReservedUids	string	UIDs that should not be used in the automatic allocation sequence.
ReservedGids	string	GIDs that should not be used in the automatic allocation sequence.
AvailableShells	string	Which shells are available in the zone. This attribute is used to populate the list of available shells that can be assigned in the user's UNIX profile.
DefaultHomeDirectory	string	The default value for the user's home directory. Typically, this attribute contains the string \${user} which is replaced with the user's login name in the home directory path. For example, if this attribute is /home/\${user}, and the user's login name is shea, the user's home directory path is defined as /home/shea.
DefaultShell	string	The default shell to use for the zone.
ZoneName	string	The name of the zone.
DefaultGid	int	The value of the GID for the group profile associated with the Active Directory group used as the default group for new users.
Description	string	The text string to use as a description for the zone. This attribute is used to display additional information about a zone in Access Manager. For example, if the value for Description attribute is Business development zone, the console displays this string after the zone name:

Logical Data Attributes for Users

The following table describes the logical attributes for the UNIX user object. Most of these attributes represent elements of the user's entry in the /etc/passwd file. The last two are specific to Delinea.

Uid	int	The value of the user's UID in the UNIX profile.
UnixName	string	The user's login name in the UNIX profile.
Gid	int	The value of the user's primary group identifier (GID).
Home	string	The path to use for the user's home directory in the user's UNIX profile.
Shell	string	The path to use for the user's shell in the user's UNIX profile.
UnixEnabled	bool	Whether the user is allowed to log on to computers in classic zones. This attribute allows a user profile to exist in a classic zone but not allowed to log on to any computers. Disabling user access is particularly useful for recording who used to own a retired UID. This attribute is not used in hierarchical zones.
AppEnabled	bool	Not used. **Note** You might see this attribute if you have a legacy version of Delinea software installed in your environment.

Logical Data Attributes for Groups

The following table describes the logical attributes for the UNIX group object. Note that there is no attribute for group membership. Group membership for groups with a UNIX profile is always derived from the Active Directory group membership.

Gid	int	The value of the group identifier (GID) for the group's UNIX profile.
UnixName	string	The group name for the group's UNIX profile.

Logical Data Attributes for Computers

The following table describes the logical attributes for the UNIX computer object. The computer attributes are used to are used to provide specific details about computers joined to a zone. In most cases, you should not change these attributes. They are created automatically by the adjoin process.

cpuCount	int	The number of CPUs detected on a computer joined to the domain. This attribute was used for licensing calculations in earlier versions of Delinea software (versions 3.0.x through 4.1.x).
agentVersion	string	The version of the Delinea Agent installed on a computer joined to the domain.

Logical Data Attributes for NIS Maps

The NIS maps for a zone are stored in a NisMaps container object. The NisMaps container object is similar to the Users, Groups, or Computers container objects that contains a zone's UNIX data. Each individual NIS map object is also a container object. The name of the object and its GUID are its only attributes. Because the NIS map objects don't require any special mapping between a logical attribute name and an Active Directory attribute name, the standard Active Directory attribute names are used. The following table describes the Active Directory attributes for NIS map objects.

Common Name (cn)	string	The name of the NIS map.
Description	string	The GUID of the map type. This attribute is used by the Access Manager console. If the attribute value is not specified, the console attempts to resolve to the most appropriate map type.

Under each NIS map object, each map entry is stored as a key/value pair of classStore objects. The following table describes the Active Directory attributes for the entry objects.

Common Name (cn)	string	The unique key of the key/value pair.
Description	string	The key for the map entry.
adminDescription	string	The value for the map entry.
wwwHomePage	string	The text comment for the map entry. This attribute is only used to display the comment in Access Manager. The comment cannot be served to NIS clients.

) [tags]: # (windows api) [priority]: # (1)

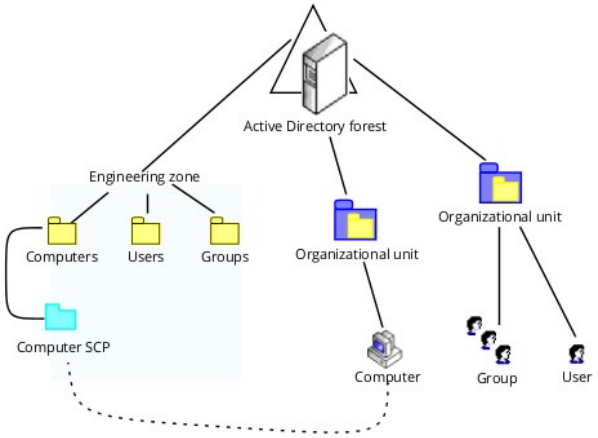
Classic SFU-Compliant Zones (version 3.5)

If you have the Microsoft Services for UNIX (SFU) schema extension installed, you have the option of using SFU-compliant zones for storing data. With SFU-compliant zones, UNIX-specific attributes for users and groups are stored in the actual Active Directory user and Active Directory group objects, using attributes in Microsoft Services For UNIX (SFU) schema extension.

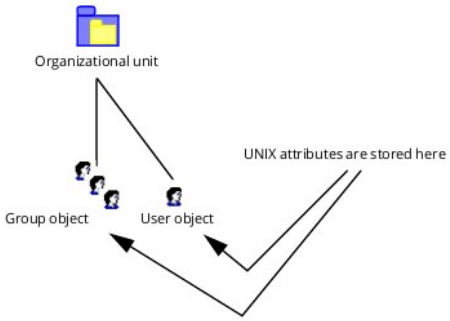
> **Note:** The schema extension must be already be installed in the forest. You cannot create SFU-compliant zones if the schema extension is not installed.

Unlike standard Delinea zones, where a single Active Directory user can have multiple UNIX profiles, a single Active Directory user can only exist in one SFU zone because there is only one set of attributes in the Active Directory user object. A single user can, however, be in any number of Delinea zones and zero or one SFU zone.

The structure of the zone and its sub-containers is the same as the classic Delinea zone layout, with each zone stored as a separate tree in the directory and sub-containers for the **Users**, **Groups**, and **Computers** in each zone, but only the Computers sub-container is used.



Unlike classic Delinea zones, in which UNIX attributes are stored in the serviceConnectionPoint objects, the SFU zones store UNIX attributes in the User and Group objects and use attributes provided by the SFU schema extension.



Zone Attributes in Classic SFU-Compliant Zones

The zone object class and its attributes in the classic SFU zone are similar to the classic Delinea zone, except that the zone must also include the NIS domain name and domain attributes. Like the classic Delinea zones, the zone object is stored as a container object, and the common name (cn) of the object must be set to the zone name. Most of the other attributes for a zone are stored as pseudo-attributes using the Active Directory description attribute.

The following table summarizes how zone attributes are stored in Active Directory for SFU-compliant zones. For more information about any attribute setting, see Zone attributes in classic Delinea zones.

ZoneName	cn:ZoneName For example: cn:default
ZoneVersion	displayName:ZoneVersion The only valid value is \$CimsZoneVersion2. For example: displayName:\$CimsZoneVersion2
Description	description:description:value For example: description:description:Pilot EMEA
NextUid	description:uidnext:value For example: description:uidnext:12098
NextGid	description:gidnext:value For example: description:gidnext:12098
ReservedUids	description:uidreserved:value For example: description:uidreserved:0-99:501
ReservedGids	description:gidreserved:value For example: description: gidreserved:1000-2500
Availableshells	description:availableshells:value For example: description:availableshells:/bin/sh
DefaultHomeDirectory	description:defaulthome:value For example: description:defaulthome:/nfs/$
DefaultShell	description:defaultshell:value For example: description:defaultshell:/bin/bash
DefaultGroup	description:defaultgid:value For example: description:defaultgid:12098
ZoneType	description:schema:Dynamic_Schema_Version The only valid value for SFU zones is: SFU_3_0 for the Microsoft Services for UNIX (SFU) versions 3.x or 4.x schema extension. For example: description:schema:SFU_3_0
NisDomain	description:Nisdomain:value This attribute describes the NIS domain for that defines the scope of the zone. For more information about this setting, see the User object attributes. For example: description: Nisdomain:XXX
SFUDomain	description:Sfudomain:value The SFU domain contains the users and groups. The members of an SFU domain can only come from one domain. For example: description:Sfudomain:mfg.ajax.org

User Attributes in Classic SFU-Compliant Zones

In classic SFU zones, UNIX-specific user attributes are stored as part of the Active Directory user object.

UnixName	MSSFU30Name:userlogin For example: MSSFU30Name:cain
Uid	MSSFU30UidNumber:value For example: MSSFU30UidNumber:458
Gid	MSSFU30GidNumber:value For example: MSSFU30GidNumber:458
Home	MSSFU30HomeDirectory:value For example: MSSFU30HomeDirectory:/home/shea
Shell	MSSFU30Shell:value For example: MSSFU30Shell:/bin/bash
NisDomain	MSSFU30NisDomain:value This attribute must be defined. Delinea uses this setting to determine if the user is a member of the zone. When you create SFU-compliant zones, you must specify the NIS domain name that should be included. For example, you can configure zone_bejing to include all users and groups whose NIS domain attribute is set to nisbejing. For example: MSSFU30NisDomain:nisbejing.local
UnixEnabled	Not supported.

Group Attributes in Classic SFU-Compliant Zones

In classic SFU-compliant zones, UNIX-specific group attributes are stored as part of the Active Directory group object.

UnixName	MSSFU30Name:GroupName For example: MSSFU30Name:performx
Gid	MSSFU30GidNumber:value For example: MSSFU30GidNumber:458
NisDomain	MSSFU30NisDomain:value This attribute must be defined. Delinea uses this setting to determine if the group is a member of the zone. When you create SFU-compliant zones, you must specify the NIS domain name that should be included. For example, you can configure zone_bejing to include all users and groups whose NIS domain attribute is set to nisbejing. For example: MSSFU30NisDomain:nisbejing.local
UnixEnabled	Not supported.

Note: The Microsoft Services for UNIX schema extension supports group membership as an attribute of the group object in the same way the RFC 2307-compliant schema does. Delinea does not use this attribute, however. Delinea uses Active Directory group membership to identify group members.

) [tags]: # (windows api) [priority]: # (1)

Classic SFU-Compliant Zones (version 4.0)

if you have the Microsoft Services for UNIX (SFU), version 4.0, schema extension installed, you have the option of using SFU-compliant zones for storing data. Delinea SFU-compliant zones for the Microsoft Services for UNIX (SFU), version 4.0, schema extension are similar to SFU-compliant zones for version 3.5, except that Microsoft Services for UNIX (SFU), version 4.0, uses the R2 schema. The UNIX-specific attributes for users and groups are still stored in the actual Active Directory user and Active Directory group objects, but use the R2 schema attributes instead of the msSFU* attributes used in Microsoft Services For UNIX (SFU), version 3.5.

> **Note:** You can only create this type of zone if the R2 schema is installed. If the R2 schema attributes are not available, you cannot create this type of zone.

Zone Attributes in Classic SFU 4.0 Zones

The zone object class and its attributes in the classic SFU-compliant zones for the Microsoft Services for UNIX (SFU), version 4.0, schema extension are the same as described in Zone attributes in classic SFU-compliant zones. For more information about any attribute setting, see Zone attributes in classic Delinea zones.

User Attributes in Classic SFU 4.0 Zones

In classic SFU-compliant zones, UNIX-specific attributes are stored as part of the Active Directory user object. With Microsoft Services for UNIX, version 4.0, however, the R2 schema attributes are used.

UnixName	uid:userlogin For example: uid:cain
Uid	uidNumber:value For example: uidNumber:458
Gid	gidNumber:value For example: gidNumber:458
Home	unixHomeDirectory:value For example: unixHomeDirectory:/home/shea
Shell	loginShell:value For example: loginShell:/bin/bash
NisDomain	MSSFU30NisDomain:value This attribute must be defined. Delinea uses this setting to determine if the user is a member of the zone. When you create SFU-compliant zones, you must specify the NIS domain name that should be included. For example, you can configure zone_bejing to include all users and groups whose NIS domain attribute is set to nisbejing. For example: MSSFU30NisDomain:nisbejing.local
UnixEnabled	Not supported.

Group Attributes in Classic SFU 4.0 Zones

In classic SFU-compliant zones, UNIX-specific attributes are stored as part of the Active Directory group object. With Microsoft Services for UNIX, version 4.0, however, the R2 schema attributes are used.

UnixName	cn:GroupName For example: cn:performx
Gid	gidNumber:value For example: gidNumber:458
NisDomain	MSSFU30NisDomain:value This attribute must be defined. Delinea uses this setting to determine if the group is a member of the zone. When you create SFU-compliant zones, you must specify the NIS domain name that should be included. For example, you can configure zone_bejing to include all users and groups whose NIS domain attribute is set to nisbejing. For example: MSSFU30NisDomain:nisbejing.local
UnixEnabled	Not supported.

Using Commands and Scripts to Perform Tasks

With an understanding of how the data is stored in Active Directory for different zones types, you can use ADEdit or LDAP commands to perform a wide range of tasks from the UNIX command line or in scripts and custom programs. The following examples illustrate how you can use the OpenLDAP command line interface (CLI) that is installed with the Delinea Agent to perform administrative tasks. The OpenLDAP CLI is Kerberos-enabled, so it is not necessary to supply credentials. All operations run with the permissions of the Active Directory user currently logged in. For information about using ADEdit to perform administrative tasks, see the *ADEdit Command Reference and Scripting Guide*.

Getting Started

The OpenLDAP commands provided with the Delinea Agent package support all of the standard command line options, plus some additional options to make it easier to use them to work with Active Directory. For example, the LDAP commands packaged with the Delinea Agent accept a URL of the form:

```
LDAP://domain_name
```

to specify the nearest domain controller in the specified domain, or a URL of:

```
`LDAP://`
```

In addition to the LDAP commands, Delinea includes several other command line programs and environment variables you may find useful in creating scripts to perform administrative tasks. For example, your scripts can take advantage of the environment variables that are set for an Active Directory user upon authentication. You can also use commands such as `adinfo` and `adfinddomain` to return information or supply input for administrative scripts.

On Linux and UNIX computers with a Delinea Agent, version 5.0 or later, you can use the ADEdit command line utility and library of commands to perform administrative tasks instead of using LDAP commands or single-purpose commands. For information about using ADEdit, see the *ADEdit Command Reference and Scripting Guide*.

Creating a Classic Zone

The following example shows the commands and data needed to create a classic Delinea zone named "zone1". Zone creation is almost identical for all zone types. Only the value of displayName and the schema pseudo-attribute differ from zone type to zone type.

Before you can create the zone itself, however, you must create an Active Directory container with the appropriate properties. The zone container must also contain four other sub-containers to accommodate the UNIX attributes for Computers, Users, Groups, and NISMaps for the zone. You can create your zone anywhere within the directory tree.

To create a zone container and zone properties using the ldapadd command:

```
ldapadd -H ldap://mydc.acme.com \<\< END_DATA

# Add the zone container
dn: cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: container
cn: zone1
description: uidnext:10005
description: gidnext:10007
description: gidreserved:0-99
description: uidreserved:0-99
description: availableshells:/bin/bash:/bin/csh:/bin/sh:/bin/tcsh
description: defaulthome:/home/\${user}
description: privategroupcreation:True
description: defaultshell:/bin/bash
description: schema:Dynamic_Schema_3_0
displayName: \\\$CimsZoneVersion2
showInAdvancedViewOnly: TRUE
name: default

# Add the Computers sub-container
dn: CN=Computers, cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: container
cn: Computers
showInAdvancedViewOnly: TRUE
name: Computers

# Add the Groups sub-container
dn: CN=Groups, cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: container
cn: Groups
showInAdvancedViewOnly: TRUE
name: Groups

# Add the Users sub-container
dn: CN=Users, cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: container
cn: Users
showInAdvancedViewOnly: TRUE
name: Users

# Add the NISMaps sub-container
dn: CN=NisMaps, cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: container
cn: NisMaps
showInAdvancedViewOnly: TRUE
name: NisMaps
END_DATA
```

Add a User to a Classic Zone

Adding a UNIX user or group profile to an Active Directory user or group object requires you to know the security identifier (SID) for the Active Directory user or Active Directory group. This information is necessary to link the UNIX attributes in the UNIX profile to its corresponding Active Directory account. One way to get this information is to use the Windows Server directory service command-line tool dsquery to return the SID for a specific user:

```
dsquery user -samid user \| dsget user -sid -samid
```

For example, to list the samAccount name and SID for the user with the samaccountname jane:

```
dsquery user -samid jane \| dsget user -sid -samid
```

> **Note:** For more information on using dsquery, search for the command on the Microsoft website.

Once you have identified the SID for a user or group, you can use the ldapadd command to add a profile for the user or group to the zone.

The following example illustrates how to add user "joe" to "zone1" where "zone1" is a classic RFC 2307-compliant zone:

```
ldapadd -H ldap://mydc.acme.com \<\< END_DATA
dn: CN=joe,CN=Users,cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: posixAccount
objectClass: serviceConnectionPoint
cn: joe
displayName: \\\$CimsUserVersion3
showInAdvancedViewOnly: TRUE
name: joe
keywords: unix_enabled:True
keywords: parentLink:S-1-5-21-397955417-626881126-188441444-512
uid: joe
uidNumber: 123
gidNumber: 234
unixHomeDirectory: /home/joe
loginShell: /bin/bash
END_DATA
```

The following example illustrates how to add the user profile "joe" to "zone1" where "zone1" is a Standard zone:

```
ldapadd -H ldap://mydc.acme.com \<\< END_DATA
dn: CN=joe,CN=Users,cn=zone1,cn=myzones,dc=acme,dc=com
objectClass: serviceConnectionPoint
cn: joe
displayName: \\\$CimsUserVersion2
showInAdvancedViewOnly: TRUE
name: joe
keywords: unix_enabled:True
keywords: parentLink:S-1-5-21-397955417-626881126-188441444-512
keywords: uid:123
keywords: gid:234
keywords: home:/home/joe
keywords: shell:/bin/bash
END_DATA
```

Basic requirements

Delinea stores UNIX-specific properties for users, groups, and computers, as well as zones and zone properties, within Active Directory by adhering to Microsoft standards for data storage. Because of this adherence to Microsoft standards, Delinea stores the UNIX-specific information differently depending on the Active Directory schema you are using and the type of Delinea zones you create.

The Delinea Windows API provides a logical abstraction of the data model so that you can manipulate Delinea-specific information without understanding the differences between zone types. If you want to manipulate the data directly without the logical abstraction, however, you need to understand the details of how UNIX-specific properties and zone information are stored for each type of zone and schema. Once you have a more detailed understanding of the physical and logical data model for each zone type, you may be able to perform tasks that are not possible with the Access Manager console.

Differences between Types of Zones

The user and group attributes described in the logical data model are stored differently in Delinea zones than they are in SFU zones.

- When you have the Microsoft Services for UNIX (SFU) schema extension, version 3.5 or version 4.0, and use SFU-compatible zones, user and group UNIX attributes are stored in the Active Directory user and Active Directory group objects.
- In classic and hierarchical Delinea and RFC 2307-compatible zones, user and group UNIX attributes are stored in one serviceConnectionPoint object per zone for each user and group.

Schemas and zones

Delinea stores UNIX identity data and Delinea zone data in Active Directory, without modifying or extending the standard Active Directory schema. Delinea stores UNIX account profiles in standard text properties in an existing Active Directory object. This data model can be used with the default Active Directory schema or with any standard schema extension provided by Microsoft. Zones and their properties are stored in the same manner, using standard text properties in an existing Active Directory object.

The default data storage model and Delinea zones enable a single Active Directory user account to be associated with any number of unique UNIX profiles that a user may have across your environment. These UNIX profiles can have unique UIDs, GIDs, home directories, and preferred shells on one or more different UNIX systems.

How the zone type can affect features

Because the details of the data model depend on the Active Directory schema and the zone type, the zone type can also impact the features a particular zone can support. For example, classic and hierarchical Delinea zones support multiple profiles for each user but SFU zones do not. The following table summarizes key differences between zone types.

Classic Delinea zones (2.x, 3.x, 4.x)	Yes	Yes	No
Classic RFC 2307compatible	Yes	Yes	No
Hierarchical Delinea zones (5.x)	Yes	Yes	Yes
Hierarchical RFC 2307-compatible zones	Yes	Yes	Yes
Hierarchical Services for UNIX (SFU) zones	No	No	Yes*
Services for UNIX (SFU), version 3.5, zones	No	No	No
Services for UNIX (SFU), version 4.0, zones	No	No	No
* A hierarchical SFU zone can only be the root parent zone. You cannot create any Service for UNIX zone as a child zone.			

For more information about differences in how data is stored in a specific zone type, see Differences between types of zones.

Supported zone types

Standard Delinea zones use the default data storage model. However, Delinea can also support the RFC 2307 data model if you are using the Microsoft RFC 2307 schema extension, or the Microsoft Services for UNIX (SFU) data model if you are using that schema extension. To support these schema extensions, you can choose the zone type you want to use for each zone.

The following table lists the supported zone types and the relationship between the zone type and the Active Directory schema.

Classic Delinea zones, versions 2.x, 3.x, and 4.x	Any schema
Hierarchical standard zones, version 5.x or later	Any schema
Classic RFC 2307-compatible zones, versions 2.x, 3.x, and 4.x	RFC 2307-compliant schema
Hierarchical RFC 2307compatible zones, version 5.x or later	RFC 2307-compliant schema
SFU-compatible zones, version 3.5	Microsoft Services for UNIX (SFU), version 3.5
SFU-compatible zones, version 4.0	Windows Services for UNIX (SFU), version 4.00

Note: Classic RFC 2307-compatible zones require Active Directory Dynamic Auxiliary Classes. The forest functional level must be at least Windows Server 2003 to use Dynamic Auxiliary Classes. All hierarchical zones require the domain functional level to be at least Windows Server 2003.

Delinea

Delinea SDK

The Delinea Software Development Kit (SDK) consists of the following:

- Application programming interfaces that packaged in a dynamic link library (.DLL).
- A compiled help file that includes complete object reference information.
- Sample scripts in multiple languages.
- Supporting documentation and reference information for UNIX developers.

On Windows computers, you can use the API to develop your own custom applications that access or modify Delinea-specific data in Active Directory.

As part of the Delinea SDK, this guide also provides detailed information about the underlying Delinea data storage model and how attributes managed by Delinea are stored in Active Directory. This information is critical for developing programs that access or modify Delinea data but are run on UNIX computers.

Development Platform

You can use the Windows API to develop programs that manage UNIX user, group, and computer profiles; zones and zone properties; and NIS maps and NIS map entries. The methods and properties that make up the API enable you to access, create, modify, and remove information stored in Active Directory. Although you can use the Windows API to manage all of the UNIX information stored in Active Directory, including UNIX profile attributes and computer accounts, the API does not run on UNIX computers.

If you want to develop programs that run on UNIX computers to access data that's stored in Active Directory, you can use the ADEdit program (adedit) or the command line programs included with the Delinea Agent for *NIX to perform queries and updates. For example, you can use ADEdit commands in custom scripts to create zones and add, update, or remove users and groups. For detailed information about using ADEdit, see the *ADEdit Command Reference and Scripting Guide*.

You can also use OpenLDAP commands to manipulate data in Active Directory directly. The key to writing programs that use OpenLDAP or other commands is understanding how the data is stored in Active Directory and the command line options supported for each of the commands you want to use. For information about using command line programs, see the man page for the corresponding program.

Depending on the task you want to perform and the development platform you want to use, you can write scripts that manage Delinea data using either the Windows API or UNIX command line programs. For example, you can perform most provisioning-related tasks using calls to the objects, methods, and properties in the Windows API, or using common LDAP commands on UNIX.

Windows SDK

If you plan to develop programs that run on Windows computers, the Delinea SDK includes the following:

- A library of commands for access control and privilege management that run in Windows PowerShell.
- Sample scripts that use the Delinea PowerShell cmdlets to illustrate common administrative tasks.
- Dynamic link libraries that expose interfaces for working with Delinea objects and attributes stored in Active Directory.
- Sample scripts that illustrate adding and removing users, groups, and zones in VBScript, PowerShell, and .NET (C#) languages.
- An overview of the programming interface architecture, its relationship to the Access Manager console, and the object properties and methods available.
- Reference information for all object properties and methods.

For more information about using Windows PowerShell cmdlets, see the *Access Control and Privilege Management Scripting Guide*.

For more information about developing COM- or .NET-based applications, see Overview of the Delinea Windows API object model and Delinea object reference

For a more detailed understanding of how Delinea-specific data is stored by zone type, see Data storage for Delinea zones

UNIX SDK

If you plan to develop programs that run on UNIX computers, the Delinea SDK includes the following:

- The ADEdit command line application and library of procedures for scripting access control and privilege management tasks.
- Sample scripts written in ADEdit that illustrate common administrative tasks.
- Detailed information about how data is stored in Active Directory and how data storage differs by zone type.
- Examples that illustrate how to use OpenLDAP commands and other command line tools to perform common administrative tasks.

For detailed information about using ADEdit, see the *ADEdit Command Reference and Scripting Guide*. For more information about developing scripts that use LDAP commands, see Data storage for Delinea zones

Overview of the Delinea Windows API Object Model

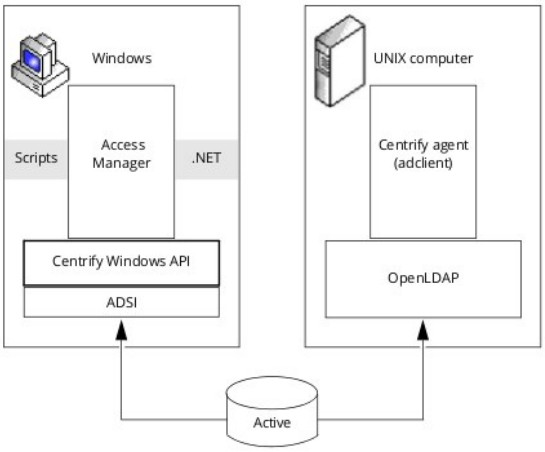
This chapter provides an overview of the architecture and capabilities of the object library exposed in the Delinea Windows API included in the SDK and how you can use the objects in applications to access and manage Delinea data on Windows computers. It includes a discussion of the Delinea framework classes, the order in which objects are created, and examples of how to use the programming interfaces in scripts written in VBScript, PowerShell, and .NET languages.

How the Delinea Windows API Relies on COM Interfaces

On Windows computers, the Delinea API supports the Component Object Model (COM) interface. The Component Object Model (COM) interface enables you to create objects that can interact with Active Directory or be used in other applications. These are re-usable objects that can provide access to all of the Delinea data stored in Active Directory. The objects can be used in any program written in .NET or COM-enabled languages. You can, therefore, create or modify applications to use these objects in COM-aware languages such as VBScript and PowerShell or .NET-compliant languages such as C#. The object model used to access the data is the same, but the specific syntax required depends on the programming language you choose to use.

The objects that make up the Delinea Windows API rely on the underlying interfaces provided by Microsoft's Active Directory Service Interfaces (ADSI). ADSI provides the base-level functions that permit applications to read and write data in Active Directory. The purpose of the Delinea Windows API is to provide a higher level of abstraction for performing Delinea-specific tasks than would be available if you were to call ADSI functions directly.

The following figure illustrates how the Delinea Windows API provides a layer of abstraction between the raw ADSI functions and the Access Manager console and other applications.



The Active Directory schema defines how all of the objects and attributes in the database are stored. When you add Delinea data to the Active Directory database, how that data is stored depends on the Active Directory schema you have installed. The Delinea Windows API, however, provides a logical view of the data, eliminating the need to know the details of how data is stored in different schemas when performing common administrative tasks. The Delinea Windows API also provides a simpler interface for accessing the well-defined set of UNIX objects that must be operated on than that offered by the general purpose ADSI. In fact, when you perform administrative tasks with the Access Manager console MMC snap-in, the console uses the same Delinea Windows API objects documented in this guide to manipulate the data.

Therefore, with the Delinea Windows API and any commonly-used Windows programming language, you can write scripts or programs that perform a wide range of tasks using Delinea data, including programs that automatically create and manage Delinea zones or update user, group, or computer properties.

> **Note**: You can use ADSI directly instead of using the Delinea Windows API, if you prefer. For more detailed information about the objects and attributes used in Active Directory when different schemas are used, see Data Storage for Delinea Zones.

Administrative Tasks You Can Perform

Using the Delinea Windows API, you can perform a wide range of common administrative tasks, including the following:

- Add, modify, and delete zones, including hierarchical zones.
- Add, modify, and delete users within zones.
- Add, modify, and delete groups in zones.
- Add, modify, and delete computers in zones.
- Create, modify, and delete rights, roles, and role assignments.
- Check on licenses and keys.
- Create, modify, and delete NIS maps and NIS map entries.

For examples of performing common activities such as these using Delinea objects, see the sample scripts and help included with the software package.

Although you can use the Delinea Windows API to perform many common administrative tasks programmatically, you cannot create new Active Directory users or groups directly. To create new Active Directory user or group objects, you must use the underlying Active Directory Service Interfaces (ADSI) rather than the abstracted interface provided by the Delinea Windows API. To learn how to create user or group objects programmatically or perform other tasks that are not provided by the Delinea Windows API, such as changing access rights, refer to Microsoft's ADSI documentation.

Delinea-specific Objects Classes

The Delinea Windows API consists of several common, interdependent classes that correspond with the core elements of Delinea-managed data, such as computers, users, groups, and zones. These basic classes provide properties, methods, and attributes that you can manipulate in programs and scripts to set or retrieve data.

The following table lists the classes that compose the Delinea Windows API.

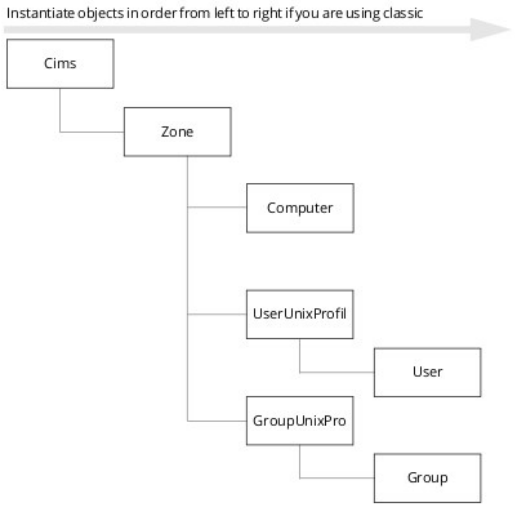
AzRoleAssignment	Represents a computer-role assignment.
Cims	Initiates interaction with Active Directory. This top-level class connects to Active Directory and prepares the Active Directory domain and forest for working with Delinea objects.
Command : Right	Represents the right to run a command, including which users and groups have that right.
Commands	Represents a collection of command rights.
Computer	Manages an individual computer account object.
ComputerGroupUnixProfiles : GroupUnixProfiles	Represents the groups in a computer zone.
ComputerRole	Manages a computer role.
ComputerRoles	Represents a collection of computer roles.
Computers	Represents a collection of computers in a zone.
ComputerUserUnixProfiles: UserUnixProfiles	Represents the users in a computer zone.
Group	Manages an individual group account object.
GroupUnixProfile	Manages the properties in the UNIX profile of a group.
GroupUnixProfiles	Represents a collection of UNIX groups in a zone.
HierarchicalGroup : GroupUnixProfile	Manages the properties in the UNIX profile of a group in a hierarchical zone.
HierarchicalUser : UserUnixProfile	Manages the properties in the UNIX profile of a user in a hierarchical zone.
HierarchicalZone : Zone	Represents a hierarchical zone.
HierarchicalZoneComputer: Computer	Manages the properties in the profile of a computer object joined to a hierarchical zone.
HzRoleAssignment : RoleAssignment	Manages a zone-level role assignment in a hierarchical zone.
InheritedRoleAsg	Represents an inherited role assignment.
Key	Represents a license key.
Keys	Represents a collection of Delinea license keys.

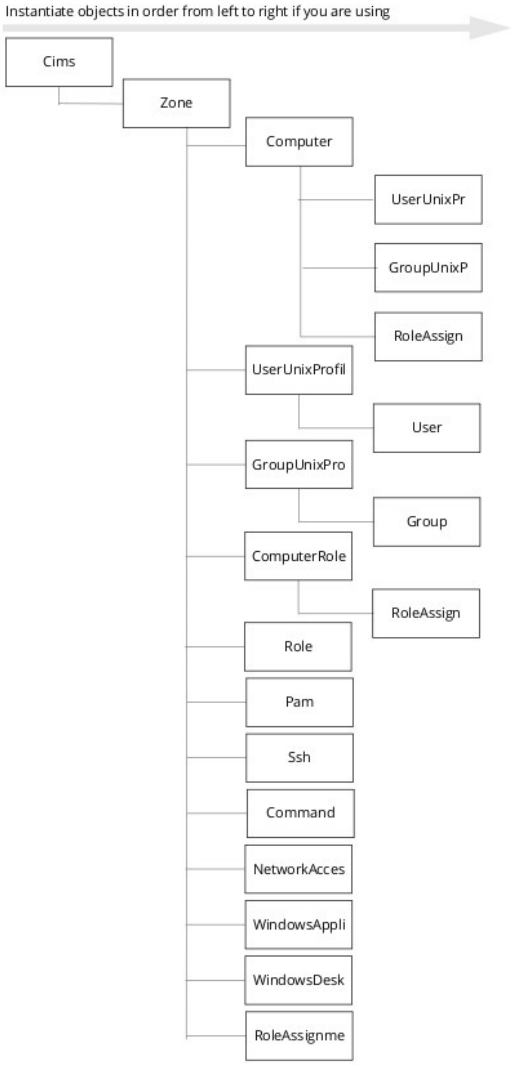
License	Represents a Delinea license.
Licenses	Represents a collection of Delinea licenses in a license container object.
LicensesCollection	Manages all the Delinea licenses in all of the Licenses parent containers defined for a forest.
MzRoleAssignment : RoleAssignment	Represents a computer-level role assignment.
NetworkAccesses	Represents a collection of network access rights.
Pam : Right	Represents a PAM (Pluggable Authentication Module) application right.
Pams	Represents a collection of PAM application rights.
Right	The base class for all rights.
Right : NetworkAccess	Represents a Windows network access right.
Right : WindowsApplication	Represents a Windows application right.
Right : WindowsDesktop	Represents a Windows desktop right.
Role	Manages a Delinea role.
RoleAssignment	Represents a role assignment.
RoleAssignments	Represents a collection of role assignments.
Roles	Represents a collection of roles.
User	Represents an individual user account object.
UserUnixProfile	Manages the properties in the profile associated with an individual UNIX user.
UserUnixProfiles	Represents a collection of users in a zone.
WindowsApplications	Represents a collection of Windows application rights.
WindowsDesktops	Represents a collection of Windows desktop rights.
WindowsUser	Represents a Windows user.
WindowsUsers	Represents a collection of Windows users.
Zone	Represents a Delinea zone, including the users, groups, and computers that have been added to the zone.

In addition to these objects, there are optional objects for managing and manipulating NIS maps and NIS map entries in Active Directory. For an overview of those objects, see Working With NIS Maps. For more information about all of the objects that enable you to manipulate Delinea-specific data in Active Directory, see Delinea Object Reference.

Creating Objects in the Proper Order

Delinea objects must be created in a particular order because some objects rely on the existence of others. For example, your application must create the Cims object first to establish communication with the Active Directory domain controller. After you create an instance of the Cims object, you must create the Zone object before you can create User, Group, or Computer objects because these objects exist in Active Directory in the context of the zone. The following figures illustrate the order for creating Delinea-related objects:





Getting and setting object properties

You can read or write most object properties; however a few are read-only. The syntax line in the object reference indicates whether an object property's value can be read ({get;}) or both read and written ({get; set;}). For example, the nextAvailableUID property can be both read and written:

int nextAvailableUID {get; set;}

To retrieve the existing value for this property, you could include a line similar to this:

read_uid_value = zone.nextAvailableUID

To set a new value for this property, you could include a line similar to this:

zone.nextAvailableUID = set_uid_value

Interface naming conventions

The Delinea Windows API objects are stored in Active Directory using the IADs interface. The IADs interface is a Microsoft standard that defines basic object features—such as properties and methods—of any Active Directory Service Interface (ADSI) object. The most common ADSI objects include users, computers, services, file systems, and file service operations. The IADs interface ensures that all ADSI objects provide a simple and consistent representation of underlying directory services.

In addition to the basic ADSI objects, Delinea-specific objects are implemented as IADs interfaces. Using interfaces for the Delinea objects enables them to change internally without requiring any changes to the API. By convention, when objects are implemented as interfaces rather than class objects, they are identified by a capital "I" as a prefix. The Delinea-specific objects that are implemented as interface objects have the same names as the classes in Delinea-specific objects classes, with the addition of the "I" prefix; for example, the IZone interface object corresponds to the Zone class.

For more information about the IADs interface and working with interface objects, see the Microsoft Developer Network Library.

Creating the Top-level Cims Object

The Cims object is the top-level object in the Delinea Windows API. This object is used to establish the connection with Active Directory and set up the environment so that other operations can be performed. Before you can retrieve any information from Active Directory, such as a zone object or user profile, you must create a Cims object. If you are writing COM-based for Delinea software, version 5.0 or later, the top-level Cims object is named Cims3. For example:

```
set cdc = CreateObject("Centrify.DirectControl.Cims3")
```

If you have scripts created for a previous version of Delinea software, you should modify the object created to be a Cims3 object.

If you are writing programs using a .NET language, the namespace for the top-level Cims object is Centrify.DirectControl.API.Cims, regardless of the version of Delinea software you are using. For example, to create the top-level Cims object in a .NET program, you would type:

```
Centrify.DirectControl.API.Cims cdc = new Centrify.DirectControl.API.Cims();
```

After creating the top-level Cims object, you can use the other Delinea objects to access and manage zones, computers, user UNIX profiles, and group UNIX profiles (see Creating Objects in the Proper Order). By writing scripts that retrieve or set object properties, you can provision users programmatically without using the Access Manager console or other MMC snap-ins.

Working With NIS Maps

In addition to the zone, computer, user, and group objects, you can use the Delinea Windows API to manage Network Information Service (NIS) maps in Active Directory. NIS maps store network information that can be used to respond to client requests on computers where the Delinea Agent cannot be installed. You can create or import NIS maps using the Access Manager console or programmatically using the API. The NIS maps you create or import are zone-specific information in Active Directory. Once the information is stored in Active Directory, NIS clients can send requests to the Delinea Network Information Service (adnisd) to receive the data.

For more detailed information about working with NIS maps and the Delinea Network Information Service, see the *Planning and Deployment Guide* and the *NIS Administrator's Guide*.

The Delinea Windows API for working with NIS maps includes the following classes:

Store	Attaches to a zone and creates NIS maps in the zone.
Map	Works with an individual map and its records.
Entry	Manages the fields in an individual record.

Writing Scripts that Use Delinea Windows API Calls

To handle Delinea tasks programmatically, you can write programs that call Delinea Windows API functions using any of the tools commonly used to write programs for Windows-based operating environments. Some of the most common of these tools include VBScript, PowerShell, and Visual Studio (C#).

To illustrate using these tools, the following sections describe how to create and run a program that uses the Delinea objects to open a zone and lists all the users in it using VBScript and PowerShell. For more detailed examples of performing common tasks using these scripting languages, see the sample scripts included in the SDK package.

- Using VBScript
- Using PowerShell
- Using Visual Studio C#

Using VBScript

In most cases, you can use VBScript to write scripts that call the Delinea Windows API.

The following steps illustrate how to create and run a VBScript script that uses the Delinea Windows API. This sample script opens a zone and lists all the users in it.

1. Verify that the computer you are using has Access Manager console or the Delinea Windows API Runtime environment from the Delinea SDK installed.
2. Verify that the computer you are using is a member of the Active Directory domain you want to work with.
3. Log in as a domain user with permission to read the zone data for the zone you will be listing.

   If you can list the users in the zone using the Access Manager console with the credentials provided, you have the correct permissions. For information about configuring a user's rights to read zone data, see the *Planning and Deployment Guide*.
4. Use a text editor to create a file called zone-list.vbs.
5. Add the following text to zone-list.vbs, replacing the domain_name and the path to the zone with a domain name and zone location appropriate for your environment.

   ```
   set cims = CreateObject("Centrify.DirectControl.Cims3")
   set zone = cims.GetZone("domain_name/zone_path/zone_name")
   set users = zone.GetUserUnixProfiles()

   for each user in users
   if (user.IsNameDefined) then
   name = user.Name
   else
   name = "<Empty>"
   end if

   if (user.IsUidDefined) then
   uid = user.Uid
   else
   uid = "<Empty>"
   end if

   wscript.echo name & " | " & uid
   next
   ```

   For example if you are using the domain test.acme.com and want to list users in the "default" zone in its default container location:

   ```
   set zone = cims.getzone("test.acme.com/program data/centrify/zones/default")
   for each user in users
   wscript.echo user.name, user.Uid
   next
   ```

6. Click **Start > Run**, then type cmd to open a command window.
7. Change directory to the location of the VBScript file and type:

   cscript zone-list.vbs

   You should see output similar to the following:

   ```
   C:\>cscript zone-list.vbs
   Microsoft (R) Windows Script Host Version 5.6
   Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

   jane 10000
   jim.smit 10002
   jimsmith 10003
   joe 10004
   paul 10006
   rachel 10016
   ```

Using PowerShell

Delinea provides a separate Access Module for PowerShell that includes predefined "cmdlets" for performing a broad range of administrative tasks without requiring any knowledge of the underlying API calls. If you prefer, however, you can write PowerShell scripts that call the Delinea Windows API directly. The following steps illustrate how to create and run a sample script that opens a zone and lists all the users in it.

1. Verify that the computer you are using has Access Manager or the Delinea Windows API Runtime environment from the Delinea SDK installed.
2. Verify that the computer you are using is a member of the Active Directory domain you want to work with.
3. Log in as a domain user with permission to read the zone data for the zone you will be listing.

   If you can list the users in the zone using Access Manager with the credentials provided, you have the correct permissions. For information about configuring a user's rights to read zone data, see the *Planning and Deployment Guide*.
4. Use a text editor to open the sample script file util.ps1.
5. Modify the util.ps1 script to specify a user name and password with administrative access to the Active Directory domain.

   For example, replace the "******" string with an administrator user name and password:

   ```
   $usrname = "administrator";
   $passwd = "1234abcepassword";
   ```
6. Use a text editor to create a file called zone-list.ps1.
7. Add the following text to zone-list.ps1, replacing the domain_name and the path to the zone with a domain controller and zone location appropriate for your environment.

   ```
   $api = "Centrify.DirectControl.API.{0}";
   $cims = New-Object($api -f "Cims");
   $objZone = $cims.GetZone("domain_name/zone_path/zone_name");
   $users = $objZone.GetUserUnixProfiles();

   foreach ($user in $users)
   {
   if ($objZone.IsHierarchical)
   {
   if ($user.IsNameDefined)
   {
   $name = $user.Name;
   }
   else
   {
   ```

```
$name = "<Empty>";
}
if ($user.IsUidDefined)
{
$uid = $user.UID;
}
else
{
$uid = "<Empty>";
}
}
else
{
$name = $user.Name;
$uid = $user.UID;
}

write-Host ("{0} I {1}" -f $name, $uid);
}
```

For example if you are using the domain test.acme.com and want to list users in the "global" zone in its default container location:

```
var zone = cims.getzone("test.acme.com/program data/centrify/zones/global");
```

8. Click **Start > Run**, then type cmd to open a command window.
9. Change directory to the location of the script file and type the following to run the script using Windows Script Host:

   ```
   cscript zone-list.ps1
   ```

   You should see output similar to the output for the VBScript sample script. For information about using the Access Module for PowerShell instead of writing scripts that call the Delinea Windows API, see the *Access Control and Privilege Management Scripting Guide*.

Using Visual Studio C#

The following steps describe how to call the Delinea Windows API when using Visual Studio 2010. Alternatively you can use the command line compilers that come in Microsoft .Net Framework SDK or the Visual Studio Express Edition. The example below is created using C#, however using **vb.net** is very similar.

Note that the .NET assemblies are not installed in the Global Assembly Cache, but they do have version numbers on them. This means that the calling applications are tied to using the same assembly versions they were compiled with. To avoid problems using the assemblies, you should install the assemblies and the applications that use the assemblies in the same directory.

1. Verify that the computer you are using has Access Manager or the Delinea Windows API Runtime environment from the Delinea SDK installed.
2. Verify that the computer you are using is a member of the Active Directory domain you want to work with.
3. Log in as a domain user with permission to read the zone data for the zone you will be listing.

   If you can list the users in the zone using Access Manager with the credentials provided, you have the correct permissions. For information about configuring a user's rights to read zone data, see the *Planning and Deployment Guide*.
4. Start **vs2010** and start a new project of type **C# console application**.
5. Click **Project > Add reference**.
6. Click the **.NET** tab, then click **Browse**.
7. Navigate to the directory where Access Manager or the SDK is installed. For example, browse to the default location C:\Program Files\Centrify\.
8. Select the following dynamic link libraries to add:

   ```
   centrifydc.api.dll
   interface.dll
   nismap.api.dll
   ```

PropSheetHost.dll
util.dll

9. Add a reference to **system.directory** services. From the **Project** menu, select **Add references**. In the **.NET** tab scroll down to system.directoryservices.dll.

10. Open the class file that contains the application's Main function. By default, Visual Studio creates this file as class1.cs.

11. Add the following code in the **Main** function, replacing the domain_name and the path to the zone with a domain controller and zone location appropriate for your environment:

```
Centrify.DirectControl.API.Cims cims = new
Centrify.DirectControl.API.Cims();
Centrify.DirectControl.API.IZone zone =
cims.GetZone("domain_name/zone_path/zone_name");
foreach (Centrify.DirectControl.API.IUserUnixProfile user in zone.GetUserUnixProfiles())
{
string name, uid;
if (zone.IsHierarchical &&
!
((Centrify.DirectControl.API.CDC50.UserUnixProfile)user).IsNameDefined)
{
name = "<Empty>";
}
else

if (zone.IsHierarchical &&
!
((Centrify.DirectControl.API.CDC50.UserUnixProfile)user).IsUidDefined)
{
uid = "<Empty>";
}
else

Console.WriteLine(name + " I " + uid);
}
```

For example if you are using the domain dc2k.seattle.test and want to list users in the "default" zone in its default container location:

```
Centrify.DirectControl.API.IZone zone =
cims.GetZone("dc2k.seattle.test/program data/centrify/zones/default");
```

12. Press **F5** to compile and run the application.

Delinea Object Reference

This chapter describes the classes, methods, and properties available for working with objects for access control and privilege management. The primary classes for working with data objects are defined in the Centrify.DirectControl.API namespace and consist of:

- AzRoleAssignment
- Cims
- Command
- Commands
- Computer
- ComputerGroupUnixProfiles
- ComputerRole
- ComputerRoles
- Computers
- ComputerUserUnixProfiles
- CustomAttributeContainer
- CustomAttributes
- CustomAttribute
- Group
- GroupUnixProfile
- GroupUnixProfiles
- HierarchicalGroup
- HierarchicalUser
- HierarchicalZone
- HierarchicalZoneComputer
- HzRoleAssignment
- InheritedRoleAsg
- Key
- Keys
- License
- Licenses
- LicensesCollection
- MzRoleAssignment
- NetworkAccess
- NetworkAccesses
- Pam

- Pams
- Right
- Role
- RoleAssignment
- RoleAssignments
- Roles
- Ssh
- Sshs
- User
- UserUnixProfile
- UserUnixProfiles
- WindowsApplication
- WindowsApplicationCriteria
- WindowsApplications
- WindowsDesktop
- WindowsDesktops
- WindowsUser
- WindowsUsers
- Zone

In addition to the basic classes, the following classes are defined in the Centrify.DirectControl.NISMap.API namespace for working with NIS maps:

- Entry
- Map
- Store

There are also separate classes for pending import groups and users. The following classes are defined in the Centrify.DirectControl.API.Import namespace for working with groups and users imported from UNIX with the “Import from UNIX” wizard:

- GroupInfo
- GroupInfos
- GroupMember
- GroupMembers
- UserInfo
- UserInfos

Delinea

AzRoleAssignment

The AzRoleAssignment class represents a computer role assignment, where a role assignment object contains information about an Active Directory object (trustee—that is, user or group) that has been added to a computer role.

Syntax

```
public interface IAzRoleAssignment : IRoleAssignment
```

Methods

The AzRoleAssignment class provides the following methods:

Commit	Commits changes in the role to Active Directory. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
ClearCustomAttributes	VBScript interface to clear the custom attributes for this class. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
Delete	Deletes the role. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
GetComputerRole	Returns the computer role that logically contains this role assignment. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
GetTrustee	Returns the trustee being assigned. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
ICustomAttributeContainer GetCustomAttributeContainer	.NET interface that returns the directory entry for the parent container object for the custom attributes for this class. (Inherited from RoleAssignment.)
SetCustomAttribute	VBScript interface to set the custom attributes for this class. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
Validate	Validates this role assignment. (Inherited from [RoleAssignment] (../roleassignment/index.md).)

Properties

The AzRoleAssignment class provides the following properties:

CustomAttributes	VBScript only: Gets or sets custom attributes for this class. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
EndTime	Determines the time at which this role becomes inactive. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
Id	Gets the GUID of the role assignment. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
IsRoleOrphaned	Indicates whether the role assignment is orphaned due to missing or invalid data. (Inherited from RoleAssignment
IsTrusteeOrphaned	Indicates whether the role assignment is orphaned due to a missing trustee. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
LocalTrustee	Gets the local trustee being assigned. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
Role	Gets the role the trustee is assigned to. (Inherited from [RoleAssignment] (../roleassignment/index.md).)

StartTime	Specifies the time from which this role becomes effective. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
TrusteeDn	Gets the distinguished name of the trustee assigned the role. (Inherited from [RoleAssignment] (../roleassignment/index.md).)
TrusteeType	Gets the trustee type of the role assignment. (Inherited from RoleAssignment.)

Discussion

A computer role describes the intended use of a group of computers; for example, the set of computers dedicated as database servers. See ComputerRoles for a discussion of computer roles.

GetComputerRole

Gets the computer role that logically contains this role assignment.

Syntax

```
IComputerRole GetComputerRole()
```

Return value

The ComputerRole instance containing this role assignment.

Discussion

The role assignment contains information about an Active Directory object that has been assigned to a computer role.

Exceptions

GetComputerRole throws an ApplicationException if no computer role is found, multiple computer roles are found, or an error occurs when accessing Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
IComputerRole compRole = objZone.GetComputerRole(strName);
if (compRole != null)
{
    Console.WriteLine("Computer role " + strName + " already exists.");
}
...
```

Cims

The Cims class is the top-level class in the Delinea Windows API.

Syntax

```
public interface ICims
```

Discussion

This class is used to establish the connection with Active Directory and set up the environment so that other operations can be performed. Before you can retrieve any information from Active Directory, you must create a Cims object. For example:

```
'Create a CIMS object to interact with Active Directory

set cims = CreateObject("Centrify.DirectControl.Cims")
```

If you are writing programs using Delinea, version 5.0 or later, the top-level Cims object is named Cims3. For example:

```
set cdc = CreateObject("Centrify.DirectControl.Cims3")
```

If you have scripts created for a previous version of Delinea software, you should modify the object created to be a Cims3 object to work with version 5.0 or later.

If you are writing programs using a .NET language, the namespace for the top-level Cims object is Centrify.DirectControl.API.Cims, regardless of the version of Delinea you are using. For example, to create the top-level Cims object in a .NET program, type:

```
Centrify.DirectControl.API.Cims cdc = new Centrify.DirectControl.API.Cims();
```

Methods

The Cims class provides the following methods:

AddComputer	Adds a computer object to a specific zone.
AddComputerZone	Adds a computer zone to a computer object.
AddWindowsComputer	Adds a Windows computer object to a hierarchical zone.
ConfigureForest	Configures the Active Directory forest to work with Delinea software.
Connect	Connects to an Active Directory domain controller.
CreateZone	Creates an individual zone object in a parent container object.
CreateZoneWithSchema	Creates an individual zone object with a specified schema type in a parent container object.
GetComputer	Returns a computer object with its related data by its directory object.
GetComputerByComputerZone	Returns a computer object given the LDAP path to the computer zone.
GetComputerByPath	Returns a computer object given the LDAP path to the computer.
GetGroup	Returns a group object with its related data by its directory object.
GetGroupByPath	Returns a group object with its related data by its LDAP path.
GetUser	Returns a user object with its related data by its directory object.

[GetUserByPath](dev/windows-api/object-reference/cims/	Returns a user object with its related data by its LDAP path.
GetWindowsUser	Returns a Windows user object.
GetWindowsUserByPath	Returns a Windows user object given the path to the object.
GetZone	Returns a zone object with its related data by object name.
GetZoneByPath	Returns a zone object with its related data by its LDAP path.
IsForestConfigured	Checks whether the forest is properly configured with valid Delinea licenses.
LoadLicenses	Returns all of the Delinea licenses for the connected domain.

Properties

The Cims class provides the following properties:

Password	Gets the password used to establish the connection to the Active Directory domain.
Server	Gets the domain controller computer name used to establish the connection to the Active Directory domain.
UserName	Gets the user name used to establish the connection to the Active Directory domain.

AddComputer

Adds a computer object to a specific zone.

Syntax

```
IComputer AddComputer(IADs computer, IZone zone)
IComputer AddComputer(string computerDn, IZone zone)
```

Parameters

Specify one of the following parameters when using this method.

computer	The Active Directory computer object that you wish to add to the zone.
computerDn	The distinguished name of the computer object.

Specify the following parameter when using this method.

zone	The zone to which you wish to add the computer object.

Return value

The newly-added computer object.

Exceptions

`AddComputer` throws an `ArgumentNullException` if any parameter value is `null` or empty.

AddComputerZone

Adds a computer zone to a computer object.

Syntax

```
IHierarchicalZoneComputer AddComputerZone(string dnsname, IZone zone)
```

Parameters

Specify the following parameters when using this method.

dnsname	The DNS host name of the Active Directory computer object to which you wish to add a computer zone.
zone	The hierarchical zone to which the computer object belongs.

Return value

The hierarchical computer object that contains the computer zone.

Discussion

Computer-level overrides for user, group, or computer role assignments are contained in a *computer zone*, a Delinea zone in Active Directory that contains properties that are specific to only one computer. Computer zones are not exposed in Access Manager.

This method adds a computer zone to a computer object in a hierarchical zone. If the Active Directory computer object exists, the method adds the computer zone to that computer. If the computer object does not exist, the method creates an orphan computer zone. When you create an Active Directory computer with the same DNS host name and call the AddComputer method to add it to the zone, this computer zone is linked to that computer object.

Exceptions

`AddComputerZone` may throw one of the following exceptions:

- `ArgumentNullException` if the DNS name parameter value is `null`.
- `ArgumentException` if the DNS name is not valid or the zone is not recognized.

AddWindowsComputer

Adds a Windows computer object to a hierarchical zone.

Syntax

```
IComputer AddWindowsComputer(IADs computer, IHierarchicalZone zone)
IComputer AddWindowsComputer(string computerDn, IHierarchicalZone zone)
```

Parameters

Specify one of the following parameters when using this method.

computer	The Active Directory computer object that you wish to add to the zone.
computerDn	The distinguished name of the computer object.

Specify the following parameter when using this method.

zone	The zone to which you wish to add the computer object.

Return value

The newly-added computer object.

Exceptions

AddWindowsComputer throws an ArgumentNullException if any parameter value is null or empty.

ConfigureForest

Configures the Active Directory forest to work with Delinea software.

Syntax

void ConfigureForest(string licenseContainerPath)

Parameters

Specify the following parameter when using this method.

licenseContainerPath	The LDAP path to the license container holding your Delinea licenses.

Discussion

Your Delinea license container must be set up before calling this function. See the Licenses class for more information about license containers.

Exceptions

ConfigureForest throws an ArgumentException if the parameter value is null or empty or if the method cannot find the license container object.

Connect

Establishes a connection to an Active Directory domain controller.

Syntax

void Connect(string server, string username, string password)

Parameters

Specify the following parameters when using this method.

server	The name of the Active Directory domain controller to which you are establishing a connection.
username	The Active Directory user account for connecting to the domain controller. The rights associated with this account used to establish the connection to Active Directory can control the operations you are allowed to perform in a script.
password	The password for the Active Directory user account connecting to the domain controller.

Discussion

This method enables you to connect to a specific domain controller using a specific user name and password, if the Active Directory server name, user name, and user password are all valid. This method is not required when you connect to Active Directory using the credentials you used to log on to the computer.

Call Cims.Connect("domaincontroller", NULL, NULL) to use the default user account

Example

The following code sample illustrates using this method in a script:

```
...
'Specify credentials to use for connecting to Active Directory
cims.Connect("ginger.ajax.org", "dane", "Niles9!");
```

CreateZone

Creates a zone in the specified parent container and returns the zone object created.

Syntax

```
IZone CreateZone(IADs container, string name)
```

Parameters

Specify the following parameters when using this method.

container	The IADs interface of the parent container object to be used to store the new zone. You can use the standard ADSI GetObject function to retrieve this interface.
name	The name of the new zone.

Return value

The zone object and its related data as Centrify.DirectControl.API.IZone.

Discussion

The CreateZone function requires you to specify the Active Directory container object or organizational unit where the zone should be created. You can use the Active Directory GetObject function to retrieve the ADSI pointer to the specified container.

Exceptions

CreateZone may throw one of the following exceptions:

- ArgumentNullException if the container object is a null reference.
- ArgumentException if the zone name is invalid.
- ApplicationException if a global catalog server error occurs.
- UnauthorizedAccessException if the container object cannot be read because of insufficient permissions.
- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this method in a script to create a new hierarchical zone named Sample_Zone in the parent container Program Data/Centrify/Zones:

```
...
string strParent = "CN=zones,CN=Centrify,CN=Program Data";
string strZone = "sample_zone";
// Create a CIMS object to interact with AD.
Cims cims = new Cims();
// Note: There is no cims.connect function.
// By default, this script will use the connection to the domain controller
// and existing credentials from the computer already logged in.
// Obtain an active directory container object.
DirectoryEntry objRootDSE = new DirectoryEntry("LDAP://rootDSE");
DirectoryEntry objContainer = new DirectoryEntry("LDAP://" + strParent + "," +
objRootDSE.Properties["defaultNamingContext"].Value.ToString());
IHierarchicalZone objZone = cims.CreateZone(objContainer, strZone) as
IHierarchicalZone;
...
```

CreateZoneWithSchema

Creates a zone with a specified schema type in the specified parent container and returns the zone object created.

Syntax

```
IZone CreateZoneWithSchema(IADs container, string name, int schema, int objectType)
```

Parameters

Specify the following parameters when using this method.

container	The IADs interface of the parent container object to be used to store the new zone.
name	The name of the new zone.
schema	The schema type to use for the new zone. This parameter determines how the zone data is stored in Active Directory. For more information about the valid schema types you can specify, see Schema.
objectType	The Active Directory object type to use for the zone. The valid values are: 0 defines the zone object as a Container object. 1 defines the zone object as an Organization Unit.

Return value

The zone object as Centrify.DirectControl.API.IZone.

Discussion

The CreateZoneWithSchema function requires you to specify the Active Directory container object or organizational unit where the zone should be created. You can use the standard Active Directory GetObject function to retrieve the ADSI pointer to the specified container.

Exceptions

CreateZoneWithSchema may throw one of the following exceptions:

- ArgumentNullException if the container object is a null reference.
- ArgumentException if the zone name is invalid.
- ApplicationException if a global catalog server error occurs.
- UnauthorizedAccessException if the container object cannot be read because of insufficient permissions.
- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this method in a script to create a new classic zone named ConsumerDiv as an organization unit in the parent container ajax.org/Corporate/Zones:

```
...
'Specify the parent container location for the zone
set objContainer = GetObject("LDAP://cn=Zones,cn=Corporate, dc=ajax,dc=org")
'Create a new zone named “ConsumerDiv”
set objZone = cims.CreateZoneWithSchema(objContainer, “ConsumerDiv”, 3, 1)
...
```

The GetObject call retrieves the ADSI pointer to the specified container.

GetComputer

Returns a computer object with all of its related Delinea-specific data, including all of the Computer object's properties and methods.

Syntax

IComputer GetComputer(IADs computer)

Parameter

Specify the following parameter when using this method:

computer	The IADs interface to the computer object you want to retrieve. You can use the standard ADSI GetObject function to retrieve this interface.

Return value

The computer object as:

Centrify.DirectControl.API.IComputer

Discussion

This method returns the computer object using the IADs interface to locate the object. The IADs interface is the directory object that represents the computer in Active Directory. The IADs object is useful for retrieving Active Directory-specific information, such as the site, for a computer object.

The method returns the computer object as Centrify.DirectControl.API.IComputer. You can then use the IComputer object to retrieve Delinea-specific information, such as the version of the Delinea Agent installed on the computer.

Exceptions

GetComputer throws an ArgumentException if the computer path is null or empty.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/east_div")
'Identify the computer you want to work with
set objIADsComputer = GetObject("LDAP://CN=magnolia,
CN=Computers,DC=ajax,DC=org")
'Get the directory object for the computer
set objComputer = cims.GetComputer(objIADsComputer)
...
```

GetComputerByComputerZone

Returns a computer object with all of its related Delinea-specific data, given the path to the computer zone associated with the computer.

Syntax

```
IHierarchicalZoneComputer GetComputerByComputerZone(string zonepath)
```

Parameter

Specify the following parameter when using this method:

zonepath	The LDAP path or distinguished name of the computer zone of the computer object you want to retrieve.

Return value

The computer object as:

```
Centrify.DirectControl.API.IHierarchicalZoneComputer
```

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, the Delinea Windows API creates a *computer zone*, a Delinea zone in Active Directory that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed in Access Manager.

This method returns the computer object using the LDAP path or distinguished name of the computer zone. The LDAP path to a computer zone uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

For example, if you use the default parent location for computer accounts in the domain `arcade.com`, the LDAP path for the computer account `magnolia` is:

```
LDAP://cn=magnolia,cn=Computers,dc=arcade,dc=com
```

Exceptions

`GetComputerByComputerZone` throws an `ApplicationException` if the method cannot find the specified computer.

GetComputerByPath

Returns a computer object with all of its related Delinea-specific data, including all of the Computer object's properties and methods.

Syntax

```
IComputer GetComputerByPath(string path)
```

Parameter

Specify the following parameter when using this method:

path	The LDAP path or distinguished name of the computer object you want to retrieve.

Return value

The computer object as:

```
Centrify.DirectControl.API.IComputer
```

Discussion

This method returns the computer object using the LDAP path or distinguished name of the object. The LDAP path to a computer account uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

For example, if you use the default parent location for computer accounts in the domain `arcade.com`, the LDAP path for the computer account `magnolia` is:

```
LDAP://cn=magnolia,cn=Computers,dc=arcade,dc=com
```

The method returns the computer object as `Centrify.DirectControl.API.IComputer`.

Exceptions

`GetComputerByPath` throws an `ArgumentException` if the computer path is `null` or empty.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/east_div")
'Identify the computer you want to work with
Set objComputer = cims.GetComputerByPath("LDAP://cn=magnolia,
cn=computers,dc=ajax,dc=org")
...
```

GetGroup

Returns an Active Directory group object with all of its related Delinea-specific data.

Syntax

IGroup GetGroup(IADs group)

Parameter

Specify the following parameter when using this method:

group	The IADs interface to the group object you want to retrieve. You can use the standard ADSI GetObject function to retrieve this interface.

Return value

The group object as:

Centrify.DirectControl.API.IGroup

Discussion

This method uses the IADs interface to locate the group object. The IADs interface is the directory object that represents the group in Active Directory. The IADs object is useful for retrieving Active Directory-specific information, such as the site, for a group.

The method returns the group object as Centrify.DirectControl.API.IGroup. You can then use the IGroup object to retrieve Delinea-specific information. For example, you can use IGroup.UnixProfiles to retrieve the UNIX group profiles associated with an Active Directory group or IGroup.AddUnixProfile to add a UNIX group profile for an Active Directory group to the zone.

Exceptions

GetGroup throws an ArgumentException if the parameter is null or empty.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/east_div")
'Identify the Active Directory group you want to work with
set objIADsGroup = GetObject("LDAP://CN=IT Interns,CN=Users, DC=ajax,DC=org")
'Get the directory object for the group
set objGroup = cims.GetGroup(objIADsGroup)
...
```

GetGroupByPath

Returns an Active Directory group object with all of its related Delinea-specific data given the path to the object.

Syntax

```
IGroup GetGroupByPath(string path)
```

Parameter

Specify the following parameter when using this method:

path	The LDAP path or distinguished name of the group object you want to retrieve.

Return value

The group object as:

```
Centrify.DirectControl.API.IGroup
```

Discussion

This method returns the group object using the LDAP path or distinguished name of the object. The LDAP path to a group uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

For example, if you use the default parent location for groups in the domain `arcade.com`, the LDAP path for the IT Interns group is:

```
LDAP://cn=IT Interns,cn=Users,dc=arcade,dc=com
```

The method returns the group object as `Centrify.DirectControl.API.Group.ObjectName`.

Exceptions

`GetGroupByPath` throws an `ArgumentException` if the group path is `null` or empty.

Example

The following code sample illustrates using this method in a script:

```
...
string strParent = "CN=zones,CN=Centrify,CN=Program Data";
if (args.Length != 2)
{
    Console.WriteLine("Usage:");
    Console.WriteLine(" test_remove_group.exe
    \"cn=sample_group,ou=groups,dc=domain,dc=tld\\" \\"default\\"");
    return;
}
string strGroup = args[0];
string strZone = args[1];
// Need to obtain an active directory container object
DirectoryEntry objRootDSE = new DirectoryEntry("LDAP://rootDSE");
DirectoryEntry objContainer = new DirectoryEntry("LDAP://" + strParent + "," +
    objRootDSE.Properties["defaultNamingContext"].Value.ToString());
string strContainerDN = objContainer.Properties["DistinguishedName"].Value as string;
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Note the lack of the cims.connect function.
// By default, this application will use the connection to the domain controller
// and existing credentials from the computer already logged in.
// Get the group object
IGroup objGroup = cims.GetGroupByPath(strGroup);
// Get the zone object
IZone objZone = cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN);
...
```

GetUser

Returns an Active Directory user object with all of its related Delinea-specific data.

Syntax

```
IUser GetUser(IADs user)
```

Parameter

Specify the following parameter when using this method:

user	The IADs interface to the user object you want to retrieve.

Return value

The user object as:

`Centrify.DirectControl.API.IUser`

Discussion

This method uses the IADs interface to locate the user object. The IADs interface is the directory object that represents the user in Active Directory. The IADs object is useful for retrieving Active Directory-specific information, such as the site, for a user.

The method returns the user object as `Centrify.DirectControl.API.IUser`. You can then use the IUser object to retrieve Delinea-specific information.

Exceptions

`GetUser` throws an `ArgumentException` if the parameter is `null` or empty.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/east_div")
'Identify the Active Directory user you want to work with
Set objUser = cims.GetUser("ajax.org/Users/Jae Smith")
...
```

GetUserByPath

Returns an Active Directory user object with all of its related Delinea-specific data given the path to the object.

Syntax

```
IUser GetUserByPath(string path)
```

Parameter

Specify the following parameter when using this method:

path	The LDAP path or distinguished name of the user object you want to retrieve.

Return value

The user object as:

```
Centrify.DirectControl.API.IUser
```

Exceptions

`GetUserByPath` throws an `ArgumentException` if the computer path is `null` or empty.

Discussion

This method returns the user object using the LDAP path or distinguished name of the object. The LDAP path to a user object uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

The method returns the user object as `Centrify.DirectControl.API.User.ObjectName`. For example, if the Active Directory user account is Jae Smith and the LDAP path to the account is `CN=Jae Smith, CN=Users, DC=ajax, DC=org`, the method returns the user object as:

```
Centrify.DirectControl.API.User.Jae Smith
```

Example

The following code sample illustrates using this method in a script:

```
...
string strUser = args[0];
if (string.IsNullOrEmpty(strUser))
{
    Console.WriteLine("User DN cannot be empty.");
    return;
}
// Obtain an active directory container object
// Configure the test container
DirectoryEntry objRootDSE = new DirectoryEntry("LDAP://rootDSE")
DirectoryEntry objContainer = new DirectoryEntry("LDAP://" + strParent + "," +
    objRootDSE.Properties["defaultNamingContext"].Value.ToString());
string strContainerDN = objContainer.Properties["DistinguishedName"].Value as string;

// Create a CIMS object to interact with AD
ICims cims = new Cims();

// Note the lack of the cims.connect function.'
// By default, this application will use the connection to domain controller
// and existing credentials from the computer already logged in.

IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

IUser objUser = cims.GetUserByPath(strUser);
if (objUser == null)
{
    Console.WriteLine("User " + strUser + " does not exist.");
```

```
    return;
}
...
```

GetWindowsUser

Returns a Windows user object.

Syntax

```
IWindowsUser GetWindowsUser(IADs user)
```

Parameter

Specify the following parameter when using this method:

user	The IADs interface to the user object you want to retrieve.

Return value

The user object as:

```
Centrify.DirectControl.API.IUser
```

Exceptions

`GetWindowsUser` throws an `ArgumentException` if the parameter is `null` or empty.

Discussion

This method uses the IADs interface to locate the user object. The IADs interface is the directory object that represents the user in Active Directory. The IADs object is useful for retrieving Active Directory-specific information, such as the site, for a user.

The method returns the user object as `Centrify.DirectControl.API.IUser`. You can then use the IUser object to retrieve Delinea-specific information.

GetWindowsUserByPath

Returns a Windows user object given the path to the object.

Syntax

```
IUser GetWindowsUserByPath(string path)
```

Parameter

Specify the following parameter when using this method:

path	The LDAP path or distinguished name of the user object you want to retrieve.

Return value

The user object as:

```
Centrify.DirectControl.API.IUser
```

Exceptions

`GetWindowsUserByPath` throws an `ArgumentException` if the path is `null` or empty.

Discussion

This method returns the user object using the LDAP path or distinguished name of the object. The LDAP path to a user object uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

The method returns the user object as `Centrify.DirectControl.API.User.ObjectName`. For example, if the Active Directory user account is `Jae Smith` and the LDAP path to the account is `CN=Jae Smith, CN=Users, DC=ajax, DC=org`, the method returns the user object as:

```
Centrify.DirectControl.API.User.Jae Smith
```

GetZone

Returns a zone object with all of its related Delinea-specific data given the zone name.

Syntax

```
IZone GetZone(string zoneName)
```

Parameter

Specify the following parameter when using this method:

zoneName	The name of the individual zone object to retrieve.

Return value

If the operation is successful, GetZone returns the named zone object and its related data as:

```
Centrify.DirectControl.API.IZone
```

Discussion

This method requires the full path to the individual zone object you want to retrieve.

This method uses the Active Directory canonical name for the zone. The canonical name for the zone uses the following naming structure:

```
domain_name/container_name/[container_name...]/zone_name
```

For example, if you use the default parent location for zones, the canonical name for the “default” zone is:

```
domain_name/Program Data/Centrify/Zones/default
```

Exceptions

GetZone may throw one of the following exceptions:

- ArgumentNullException if no zoneName parameter is passed.
- ApplicationException if the specified zone name is not valid.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/east_div")
...
```

GetZoneByPath

Returns a zone object with all of its related Delinea-specific data given its LDAP path.

Syntax

```
IZone GetZoneByPath(string path)
```

Parameter

Specify the following parameter when using this method:

path	The full LDAP path to the individual zone object you want to retrieve.

Return value

If the operation is successful, GetZoneByPath returns the zone object and its related data as Centrify.DirectControl.API.IZone.

Discussion

The LDAP path to a zone uses the following format:

```
LDAP://[domain/]attr=name,[...],dc=domain_part,[...]
```

For example, if you use the default parent location for zones in the domain arcade.com, the LDAP path for the "default" zone is:

```
LDAP://cn=default,cn=zones,cn=Centrify,cn=program data, dc=arcade,dc=com
```

> **Note**: The LDAP portion of the path is case sensitive. If you are unsure of the LDAP path for a zone, you can use the adinfo command on any computer in the zone to display the path.

Exceptions

GetZoneByPath may throw one of the following exceptions:

- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- ApplicationException if the object cannot be located by the specified path.

Example

The following code sample illustrates using this method in a script:

```
...
string strUser = args[0];
if (string.IsNullOrEmpty(strUser))
{
    Console.WriteLine("User DN cannot be empty.");
    return;
}
// Obtain an active directory container object
// Configure the test container
DirectoryEntry objRootDSE = new DirectoryEntry("LDAP://rootDSE");
DirectoryEntry objContainer = new DirectoryEntry("LDAP://" + strParent + "," +
    objRootDSE.Properties["defaultNamingContext"].Value.ToString());
string strContainerDN = objContainer.Properties["DistinguishedName"].Value as string;
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Note the lack of the cims.connect function.
// By default, this application will use the connection to the domain controller
// and existing credentials from the computer already logged in.
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;
```

```
IUser objUser = cims.GetUserByPath(strUser);
if (objUser == null)
{
    Console.WriteLine("User " + strUser + " does not exist.");
    return;
}
…
```

IsForestConfigured

Indicates whether the Active Directory forest is configured with valid Delinea licenses.

Syntax

```
bool IsForestConfigured()
```

Return value

Returns `true` if the Active Directory forest is properly configured with at least one readable license, or `false` if no valid licenses are found.

Exceptions

`IsForestConfigured` may throw one of the following exceptions:

- `COMException` if there is an LDAP error. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- `ApplicationException` if there is a global catalog server error. This exception may be thrown if the global catalog is not found or if LDAP errors occur during the discovery of the global catalog.

Example

The following code sample illustrates using this method in a script:

```
…
'Check the Active Directory forest for licenses
If not cims.IsForestConfigured then
    wScript.Echo “Forest is not configured”
End if
…
```

LoadLicenses

Returns all of the Delinea licenses installed on the connected domain.

Syntax

```
ILicensesCollection LoadLicenses()
```

Return value

The Centrify.DirectControl.API.Licenses object containing the collection of Delinea licenses installed on the connected domain.

Discussion

This method returns the collection of all licenses in all of the license parent containers found in the forest and represented in the LicensesCollection object.

Exceptions

LoadLicenses throws an ApplicationException if no license container is found or if any error occurs while accessing Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
'Get the collection of licenses
If cims.IsForestConfigured = true then
    set objLicense = LoadLicenses()
end if
...
```

Password

Gets the logged-in user's password for connecting to the domain.

Syntax

```
string Password {get;}
```

Property value

The password used to connect to the domain.

Example

The following code sample illustrates using this property in a script:

```
…
'Connect to the domain controller Active Directory
cims.Connect("paris.ajax.org","pierre","lesbleUs")
‘Display the password used to log on
wScript.Echo “Current Password:” & cims.Password
…
```

Server

Gets the Active Directory domain controller name being used to establish the connection to the Active Directory domain.

Syntax

```
string Server {get;}
```

Property value

The domain controller name used to connect to the domain.

Example

The following code sample illustrates using this property in a script:

```
…
'Connect to the domain controller Active Directory
cims.Connect("paris.ajax.org","pierre","lesbleUs")
‘Display the domain controller name
wScript.Echo “Connected to: “ & cims.Server
…
```

UserName

Gets the Active Directory user name used to establish the connection to the Active Directory domain.

Syntax

```
string UserName {get;}
```

Property value

The Active Directory user name used to connect to the domain.

Example

The following code sample illustrates using this property in a script:

```
…
'Connect to the domain controller Active Directory
cims.Connect("paris.ajax.org","pierre","lesbleUs")
‘Display the user name
wScript.Echo “Current User Credentials:” cims.UserName
…
```

Command

The Command class represents a command right.

Syntax

```
public interface ICommand : IRight
```

Methods

The Command class provides the following methods:

Commit	Commits changes in the right to Active Directory. (Inherited from Right.)
Delete	Removes the right. (Inherited from Right.)

Properties

The Command class provides the following properties:

AllowNestedExecution	If true, allows the command to start another program or open a new shell.
AuthenticationType	Specifies the type of authentication required to run a command.
CommandPattern	Gets or sets the command string that is matched to identify the command.
CommandPatternType	Gets or sets the type of pattern used to match the command.
Description	Gets or sets the description of the right. (Inherited from Right.)
DzdoRunAsGroupList	Gets or sets the list of groups allowed to run this command using dzdo.
DzdoRunAsUserList	Gets or sets the list of users and groups allowed to run this command using dzdo.
DzshRunAsUser	Gets or sets the user this command runs as when executed with dzsh.
Guid	Gets the GUID of the right.
IsReadable	Indicates whether the right is readable. (Inherited from Right.)
IsResetVariables	Resets or removes a default set of environment variables when running the command.
IsWritable	Indicates whether the right is writable. (Inherited from Right.)
MatchPath	Gets or sets the path for matching the specified command name.
Name	Gets or sets the name of the right. (Inherited from Right.)
PreserveGroupMembership	Determines whether to retain the user's group membership while executing a command.
UMask	Gets or sets the UMask value to use for the command.
VariablesToAdd	Gets or sets the list of environment name-value pairs to add, such as var1=a,var2=b,var3=c.

VariablesToKeepOrDelete	Gets or sets the list of environment variables to keep or delete.
Weight	Gets or sets the weight for this command.
Zone	Gets the zone to which this right belongs. (Inherited from Right.)

Discussion

A command right controls who has permission to run a specific command in a zone.

AllowNestedExecution

Determines whether the command is allowed to start another program or open a new shell.

Syntax

```
bool AllowNestedExecution {get; set;}
```

Property value

Set to `true` if the command is allowed to start another program or open a new shell. The default is `true`.

AuthenticationType

Determines the type of authentication required to run a command.

Syntax

```
AuthenticationType AuthenticationType {get; set;}
```

Property value

The default value is to have no authentication required. If authentication is required, this property specifies the account used to authenticate before allowing use of the command right.

Possible values:

```
public enum AuthenticationType
{
    // No authentication required
    None = 0,
    // Authenticate using logged-on user password
    LoggedOnUserPassword,
    // Authenticate using target run-as user password
    RunasUserPassword
}
```

CommandPattern

Gets or sets the command string that is matched to identify the command.

Syntax

```
string CommandPattern {get; set;}
```

Property value

The path to the command string.

Discussion

Use the CommandPatternType property to get or set the type of command-pattern string matching to use.

Exceptions

CommandPattern may throw the following exception:

- ArgumentException if the command pattern value is empty or null.

Example

Glob expression: "rm ./*"

Regular expression: "!finger sjohan \l epage"

CommandPatternType

Gets or sets the type of pattern used to match the command.

Syntax

```
PatternType CommandPatternType {get; set;}
```

Property value

The type of pattern-matching to use to identify the command.

Possible values:

```
public enum PatternType
{
    // Match using glob pattern
    Glob = 0,
    // Match using regular expression
    RegularExpression = 1
}
```

DzdoRunAsGroupList

Gets or sets the list of groups allowed to run this command using dzdo.

Syntax

```
string DzdoRunAsUserList {get; set}
```

Property value

A comma-separated string of group names (for example, "group1,group2,group3"). If a value of "*" is set, any group enabled for the zone can run the command.

DzdoRunAsUserList

Gets or sets the list of users and groups allowed to run this command using dzdo.

Syntax

```
string DzdoRunAsUserList {get; set}
```

Property value

A comma-separated string of user and group names (for example, "user1,user2,group1"). If a value of "*" is set, any user enabled for the zone can run the command.

Discussion

If you don't specify a list in this property, by default only the root user can run the command.

DzshRunAsUser

Gets or sets the user under which the command runs when executed using dzsh.

Syntax

```
string DzshRunAsUser {get; set}
```

Property value

The user name of a user authorized to run the sh command.

The default value is the current user.

Guid

Gets the GUID of the command right.

Syntax

```
Guid Guid {get;}
```

Property value

The GUID of the command right.

IsResetVariables

Determines whether to reset environment variables when running the command.

Syntax

```
bool IsResetVariables {get; set;}
```

Property value

Set `true` to reset environment values when the user runs the command.

Discussion

The `dzdo.env_keep` configuration parameter in the `centrifydc.conf` file defines a set of environment variables to retain from the current user's environment when the command is run, regardless of whether the `IsResetVariables` property is `true` or `false`. When you set this property true, if you want to specify additional variables to retain from the user's environment, list the variables in the VariablesToKeepOrDelete property.

The dzdo.env_delete configuration parameter in the centrifydc.conf file defines a set of environment variables to delete from the current user's environment when the command is run, regardless of whether the IsResetVariables property is true or false. When you set this property false, if you want to specify additional environment variables to remove, list those variables in the VariablesToKeepOrDelete property.

MatchPath

Gets or sets the match type for the path of the command.

Syntax

```
string MatchPath {get; set;}
```

Property value

The default value is "USERPATH".

Possible values to match:

- "USERPATH" Standard user path, starting with a forward slash (/).
- "SYSTEMPATH" Standard system path, starting with a forward slash (/).
- "SYSTEMSEARCHPATH" System search path, starting with a forward slash (/).
- A custom specific path, starting with a forward slash (/).
- All paths using a single asterisk (*).

PreserveGroupMembership

Determines whether to retain the user's group membership while executing the command.

Syntax

```
bool PreserveGroupMembership {get; set;}
```

Property value

Set `true` to preserve group membership.

The default is `false`.

UMask

Determines the user file-creation mode mask (umask) value to use for the command.

Syntax

```
string UMask {get; set;}
```

Property value

The default Unix file permissions for new files created by the command, expressed as an octal number. For example, `764` or `077`. The default value is `077`.

Exceptions

UMask throws an `ArgumentException` if the specified permissions mask is not valid.

VariablesToAdd

Gets or sets a comma-separated list of environment variable name-value pairs to add when the command is executed.

Syntax

```
string VariablesToAdd {get; set;}
```

Property value

A comma-separated string of name-value pairs (for example, "var1=a,var2=b,var3=c"). If a value of null or empty string is set, no name-value pairs are added. The default is null.

Exceptions

VariablesToAdd throws an ArgumentException if VariablesToAdd contains an empty entry, the name, value, or name-value pair is invalid, or you listed duplicate variables.

VariablesToKeepOrDelete

Get or set a comma-separated list of environment variables to keep or delete, depending on the value of the IsResetVariables property.

Syntax

```
string VariablesToKeepOrDelete {get; set;}
```

Property value

A comma-separated list of environment variables. The default is null.

Discussion

This list is used by the IsResetVariables method to determine which variables should be kept or deleted when the command identified by this Command object is run.

Exceptions

VariablesToKeepOrDelete throws an ArgumentException if the variable name is invalid or you specified duplicate variables.

Weight

Gets or sets the weight of the command.

Syntax

```
int Weight {get; set;}
```

Property value

The command priority. The default value is `0`.

Discussion

This number is when handling multiple matches for commands specified by wild cards. If commands specified by this command object match commands specified by another command object, the command object with the higher command priority prevails. The higher the value of the `Weight` property, the higher the priority.

Commands

The Commands class manages a collection of Command objects.

Syntax

```
public interface ICommands
```

Methods

The Commands class provides the following method:

GetEnumerator	Returns an enumeration of Command objects.

GetEnumerator

Returns an enumeration of Command objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of Command objects.

Computer

The Computer class represents a Delinea-managed computer object.

Syntax

```
public interface IComputer
```

Methods

The Computer class provides the following methods:

Commit	Commits changes to the computer object and saves them in Active Directory.
Delete	Removes the computer profile from Active Directory.
GetDirectoryEntry	Returns the directory entry for a computer object.
Refresh	Reloads the computer object data from the data in Active Directory.

Properties

The Computer class provides the following properties:

AdsiInterface	Gets the IADs interface for the computer object from Active Directory.
ADsPath	Gets the LDAP path to the computer object.
AgentVersion	Gets the version number of the Delinea Agent as it is stored in Active Directory.
CanonicalName	Gets the canonical name of the computer object.
IsOrphan	Indicates whether the computer profile has a computer object associated with it.
IsReadable	Indicates whether the computer object is readable.
IsWritable	Indicates whether the computer object is writable.
JBossEnabled	Gets or sets the attribute that enables JBoss access for a computer account.
Name	Gets the computer name of the computer object.
ProfileADsPath	Gets the LDAP path to the computer profile associated with a computer object.
SchemaVersion	Gets the version of the Active Directory schema.
TomcatEnabled	Gets or sets the attribute that enables Tomcat access for a computer account.
Version	Gets the version number of the Active Directory schema.
WebLogicEnabled	Gets or sets the attribute that enables WebLogic access for a computer account.
WebSphereEnabled	Gets or sets the attribute that enables WebSphere access for a computer account

Zone	Gets the zone associated with the Computer object.
ZoneMode	Gets the zone mode of the computer.

Commit

Commits any changes or updates to the computer object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

When you use this method, it checks and validates the computer properties before saving the object in Active Directory. For example, before saving, the method validates that the computer name doesn't exceed the maximum length or contain invalid characters.

Exceptions

`Commit` may throw one of the following exceptions:

- `ApplicationException` if the changes you are attempting to save contain one or more errors.
- `COMException` if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- `InvalidOperationException` if the computer profile cannot be found. This usually indicates that the computer is not in a zone.
- `UnauthorizedAccessException` if there are insufficient permissions to commit the Active Directory computer account object.

Example

The following code sample illustrates using `Commit` for a computer object in a script:

```
...
set objZone = cims.GetZone("sierra.com/Performix/Zones/HongKong")
'Identify the computer account
Set objComp = cims.GetComputer("sierra.com/Performix/Computers/chu")
'Set a property and save the changes to the computer account
Set objComp.WebSphereEnabled = true
objComp.Commit
...
```

Delete

Removes the computer profile from Active Directory.

Syntax

```
void Delete()
```

Discussion

The computer profile is the service connection point associated with the computer object. This method does not remove the computer object itself from Active Directory.

Exceptions

`Delete` may throw one of the following exceptions:

- `InvalidOperationException` if the computer profile cannot be found. This usually indicates that the computer is not in a zone.
- `UnauthorizedAccessException` if there are insufficient permissions to delete the Active Directory computer profile.

Example

The following code sample illustrates using `Delete` for a computer object in a script:

```
...
set objZone = cims.GetZone("sierra.com/Performix/Zones/HongKong")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aix_fr03,
cn=Performix,CN=Computers,DC=sierra,DC=com")
'Delete the profile for the computer account
objComp.Delete
...
```

GetDirectoryEntry

Returns the directory entry for the computer object.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The DirectoryEntry attribute of the computer object.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Example

The following code sample illustrates using GetDirectoryEntry for a computer object in a script:

```
...
'Identify the computer account
IComputer computer = cims.GetComputerByPath("LDAP://CN=sage,
CN=Computers,DC=arcade,DC=com");
'Get the directory entry for the computer account
DirectoryEntry computerEntry = computer.GetDirectoryEntry();
'Rename the computer account
computerEntry.Rename("CN=sagebrush");
...
```

Refresh

Reloads the Computer object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the computer properties in the cached object to ensure it is synchronized with the latest information in Active Directory.

Exceptions

`Refresh` may throw one of the following exceptions:

- `InvalidOperationException` if the computer profile cannot be found. This usually indicates that the computer is not in a zone.
- `COMException` if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using `Refresh` for a computer object in a script:

```
...
set objZone = cims.GetZone("sierra.com/Performix/Zones/HongKong")
'Identify the computer account
Set objComp = cims.GetComputer("sierra.com/Performix/Computers/chu”)
'Set a property and save the changes to the computer account
Set objComp.WebSphereEnabled = true
objComp.Commit
'Refresh the computer object and display the result
objComp.Refresh
wScript.Echo objComp.WebSphereEnabled
...
```

AdsiInterface

Gets the IADs interface for the computer object from Active Directory.

Syntax

IADs AdsiInterface {get;}

Property value

The IADs interface for the computer object.

Example

The following code sample illustrates using AdsiInterface for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=Centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
'Display the LDAP path of the parent container
wScript.Echo objComp.AdsiInterface.Parent
...
```

ADsPath

Returns the LDAP path to the computer object.

Syntax

```
string ADsPath {get;}
```

Property value

The LDAP path to the computer object in the following format:

```
LDAP://CN=aixserver,CN=computers,DC=sierra,DC=com
```

Returns `null` if the computer profile is an orphan.

Example

The following code sample illustrates using `ADsPath` for a computer object in a script:

```
...
set objZone = cims.GetZone("ajax.org/UNIX/Zones/")
'Identify the computer account
Set objComp = cims.GetComputer("ajax.org/Computers/backup78")
wScript.Echo "LDAP path: " & objComp.ADsPath
...
```

AgentVersion

Gets the version number of the Delinea Agent as it is stored in Active Directory.

Syntax

```
string AgentVersion {get;}
```

Property value

The version number of the Delinea Agent.

Example

The following code sample illustrates using AgentVersion for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=Centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
wScript.Echo "Centrify Agent: " & objComp.AgentVersion
...
```

CanonicalName

Gets the Active Directory canonical name of the computer object.

Syntax

```
string CanonicalName {get;}
```

Property value

The canonical name of the computer object.

Example

The following code sample illustrates using CanonicalName for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=solaris10-dev,
CN=computers,DC=sierra,DC=com")
wScript.Echo "Canonical name: " & objComp.CanonicalName
...
```

IsOrphan

Indicates whether the Delinea profile associated with a computer object is an orphan.

Syntax

```
bool IsOrphan {get;}
```

Property value

Returns `true` if the computer profile is an orphan, or `false` if the object is not an orphan.

Discussion

A Delinea computer profile can become an orphan if the computer object it is associated with is deleted manually using Active Directory Users and Computers or ADSI. Orphan data can consume disk space and reduce performance for directory services and should be removed.

Example

The following code sample illustrates using `IsOrphan` for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Check for orphan profiles
for each computer in objZone.GetComputers
    If computer.IsOrphan then
        wScript.Echo computer.Name
    end if
next
...
```

IsReadable

Indicates whether the data associated with the computer object is readable using the current permissions.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the computer object is readable, or false if the object is not readable.

Discussion

This property returns a value of true if the user accessing the computer object in Active Directory has sufficient permissions to read its properties.

Example

The following code sample illustrates using IsReadable for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputer("ajax.org/Computers/backup78")
If not objComp.IsReadable then
    wScript.Echo “Computer account is not readable!”
end if
...
```

IsWritable

Indicates whether the data associated with the computer object is writable by the current user.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns true if the computer object is writable, or false if the object is not writable.

Discussion

This property returns a value of true if the user accessing the computer object in Active Directory has sufficient permissions to change the computer object's properties.

Example

The following code sample illustrates using IsWritable for a computer object in a script:

```
...
set objZone = cims.GetZone("ajax.org/UNIX/Zones/Pilot zone")
'Identify the computer account
Set objComp = cims.GetComputer("ajax.org/Computers/backup78")
If not objComp.IsWritable then
    wScript.Echo "You cannot save changes to this computer account!"
end if
...
```

BossEnabled

Gets or sets the attribute that indicates whether the computer is enabled as a server for JBoss applications.

Syntax

```
bool JBossEnabled {get; set;}
```

Property value

Set to true if access to JBoss applications is enabled for the computer account, or false if access to JBoss applications is not enabled.

Exceptions

JBossEnabled throws an InvalidOperationException if you try to set this property when the computer is not in a zone. For example, if you are using Delinea Express or have joined the domain using the --workstation option, you should not use this property.

Example

The following code sample illustrates using JBossEnabled for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
Set objComp.JBossEnabled = false
objComp.Commit
...
```

Name

Gets the computer name of the computer object.

Syntax

```
string Name {get;}
```

Property value

The computer name of the computer object.

Example

The following code sample illustrates using Name for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=magnolia,
CN=computers,DC=sierra,DC=com")
'Display the computer account name
wScript.Echo "Computer name: " & objComp.Name
...
```

ProfileADsPath

Gets the LDAP path to a computer object's UNIX profile.

Syntax

```
string ProfileADsPath {get;}
```

Property value

The LDAP path for the computer profile associated with the computer object.

Example

The following code sample illustrates using ProfileADsPath for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=magnolia,
CN=computers,DC=sierra,DC=com")
'Display the LDAP path to the computer's UNIX profile
wScript.Echo "LDAP path: " & objComp.ProfileADsPath
...
```

SchemaVersion

Gets the version of the Active Directory data schema.

Syntax

```
int SchemaVersion {get;}
```

Property value

The version of the schema as an integer (int).

Discussion

This property is used internally by the Delinea .NET module to identify the Active Directory schema being used.

> **Note**: This property is designed for COM-based programs that support a 32-bit signed number. The property Version can be used in place of this property in .NET programs because .NET supports 64-bit signed numbers.

Example

The following code sample illustrates using SchemaVersion for a computer object in a script:

```
...
set objZone = cims.GetZone("ajax.org/UNIX/Zones/Pilot zone
)
'Identify the computer account
Set objComp = cims.GetComputer("ajax.org/Computers/backup78")
wScript.Echo “Schema version: “ & objComp.SchemaVersion
...
```

TomcatEnabled

Determines whether the computer is enabled as a server for Tomcat applications.

Syntax

```
bool TomcatEnabled {get; set;}
```

Property value

Set to `true` if access to Tomcat applications is enabled for the computer account, or `false` if not.

Exceptions

`TomcatEnabled` throws an `InvalidOperationException` if you try to set this property when the computer is not in a zone. For example, if you are using Delinea Express or have joined the domain using the `--workstation` option, you should not use this property.

Example

The following code sample illustrates using `TomcatEnabled` for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
Set objComp.TomcatEnabled = true
objComp.Commit
...
```

Version

Gets the version number of the Active Directory data schema.

Syntax

```
long Version {get;}
```

Property value

The version number of the Active Directory schema as a long integer value.

Discussion

This property can be used only in .NET programs because .NET supports 64-bit signed numbers. This property cannot be used in COM-based programs. For COM-based programs, use the SchemaVersion property instead.

Example

The following code sample illustrates using Version for a computer object in a script:

```
...
// Create the top-level object
Cims cims = new Cims()
// Identify the computer account
IComputer computer = cims.GetComputer("ajax.org/Computers/backup78")
Console WriteLine "Schema version number: " + computer.Version
...
```

WebLogicEnabled

Determines whether the computer is enabled as a server for WebLogic applications.

Syntax

```
bool WebLogicEnabled {get; set;}
```

Property value

Set to true if access to WebLogic applications is enabled for the computer account or false if not.

Exceptions

WebLogicEnabled throws an InvalidOperationException if you try to set this property when the computer is not in a zone. For example, if you are using Delinea Express or have joined the domain using the --workstation option, you should not use this property.

Example

The following code sample illustrates using WebLogicEnabled for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
Set objComp.WebLogicEnabled = true
objComp.Commit
...
```

WebSphereEnabled

Determines whether the computer is enabled as a server for WebSphere applications.

Syntax

```
bool WebSphereEnabled {get; set;}
```

Property value

Set to true if access to WebSphere applications is enabled for the computer account, or false if not.

Exceptions

WebSphereEnabled throws an InvalidOperationException if you try to set this property when the computer is not in a zone. For example, if you are using Delinea Express or have joined the domain using the --workstation option, you should not use this property.

Example

The following code sample illustrates using WebSphereEnabled for a computer object in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
Set objComp.WebSphereEnabled = false
objComp.Commit
...
```

Zone

Gets or sets the zone object associated with the computer object.

Syntax

```
IZone Zone {get; set;}
```

Property value

The zone object for the zone of the computer account.

Discussion

Each computer object can only be associated with one zone: the zone used to join the computer to its Active Directory domain. This property gets or sets the zone object for the zone to which the computer is currently joined.

Exceptions

`Zone` may throw one of the following exceptions:

- `ApplicationException` if the zone you specify is `null`, an unsupported type, or already in use; or if you were trying to move the computer object to a new domain and the operation failed.
- `InvalidOperationException` if the computer is not zoned.

Example

The following code sample illustrates using `Zone` for a computer object in a script:

```
…
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the computer account
Set objComp = cims.GetComputerByPath("LDAP://CN=aixserver,
CN=computers,DC=sierra,DC=com")
‘Display the zone name
wScript.Echo objComp.Zone.Name
…
```

ZoneMode

Gets the zone mode of the computer; used internally by the .NET module.

Syntax

```
ComputerZoneMode ZoneMode {get;}
```

Property value

The zone mode of the computer.

Possible values:

```
public enum ComputerZoneMode
{
    // Unknown
       Unknown = 0,
    // The computer is joined to a zone
       Zoned = 1,
    // The computer is in workstation mode
       Workstation = 2,
    // The computer is in express mode
       Express = 4,
    // The computer is in null zone mode (no pam or nss)
       NullZone = 8,
};
```

ComputerGroupUnixProfiles

Enumerates groups under a computer zone.

Syntax

```
public interface IComputerGroupUnixProfiles : IGroupUnixProfiles
```

Methods

The ComputerGroupUnixProfiles class provides the following methods:

Find	Finds the group added to the computer.
GetEnumerator	Returns the enumeration of GroupUnixProfile objects. (Inherited from GroupUnixProfiles.)
Refresh	Reloads the cached GroupUnixProfile objects. (Inherited from GroupUnixProfiles.)

Properties

The ComputerGroupUnixProfiles class provides the following properties:

Count	Gets the number of GroupUnixProfile objects in this set. (Inherited from GroupUnixProfiles.)
IsEmpty	Determines whether the collection of UNIX group profiles is empty. (Inherited from GroupUnixProfiles.)

Find

Finds the group added to the computer.

Syntax

```
IHierarchicalGroup Find(IHierarchicalZoneComputer computer)
```

Parameter

Specify the following parameter when using this method:

computer	The computer to search.

Return value

The hierarchical group added to the computer, if any, or `null` if no group is found.

ComputerRole

This class represents a computer role.

Syntax

public interface IComputerRole

Methods

The ComputerRole class provides the following methods:

AddAccessGroup	Adds a user group to this computer role.
AddRoleAssignment	Adds an empty role assignment.
AddUser	Adds a user role assignment to this computer role.
ClearCustomAttributes	VBScript interface to clear the custom attributes for this class.
Commit	Saves changes.
Delete	Deletes this computer role.
GetAccessGroup	Gets a user group assigned to this computer role.
GetAccessGroups	Gets the user groups assigned to this computer role.
GetCustomAttributeContainer	Gets the directory entry for the parent container object for the custom attributes for this class.
GetGroup	Gets the AD computer group associated with this computer role.
GetRoleAssignment	Gets the role assignment for a specified role and user.
GetRoleAssignmentById	Gets the role assignment, given a GUID.
GetRoleAssignments	Returns all the user role assignments under this computer role.
GetRoleAssignmentToAllADUsers	Returns the role assignment given to all Active Directory users who have a specified role.
GetRoleAssignmentToEveryone	Returns the role assignment given to all users who have a specified role.
GetUser	Gets a user assigned to this computer role.
GetUsers	Gets the collection of users assigned to this computer role.
ICustomAttributeContainer GetCustomAttributeContainer	.NET interface that returns the directory entry for the parent container object for the custom attributes for this class.
SetCustomAttribute	VBScript interface to set the custom attributes for this class.
Validate	Validates the changes made to this computer role.

Properties

The ComputerRole class provides the following properties:

CustomAttributes	VBScript only: Gets or sets custom attributes for this computer role.
Description	Gets or sets the description of this computer role.
Group	Gets or sets the AD computer group associated with this computer role.
IsOrphan	Indicates whether this computer role is an orphan.
Name	Gets or sets the name of this computer role.
Zone	Gets the zone of this computer role.

Discussion

A computer role describes the intended use of a group of computers; for example, the set of computers dedicated as database servers. Each computer role has one associated Active Directory computer group, which identifies the computers that have that use. You can assign any number of users or user groups to a computer role, with each user or user group having the permissions necessary to perform a set of functions on computers in that computer role.

> **Note**: Although there are conceptual similarities, a computer role is not a variety of access role. Whereas an access role is a set of permissions assigned to an Active Directory user or user group, a computer role defines the intended use of a group of computers. For example, the DBServer computer role might be associated with the Active Directory DatabaseServers computer group. Two user groups might be assigned to the DBServer computer role: DBUsers, which has the DatabaseUsers access role; and DBAdmins, which has the DatabaseAdmins role.

You can add custom attributes to role definitions and role assignments. For example, you might want to use a custom attribute to reference a ticket number associated with a specific type of access request, role definition, or temporary role assignment. Custom attributes are optional and you can use them to capture any kind of information that is meaningful to your organization.

You can add custom attributes when defining or modifying a role, defining or modifying a computer role, or when modifying role assignment properties.

The key point is that you can use the field for any type of information you might find useful. Customers most often want to reference a trouble/request ticket but the field can contain whatever you want.

AddAccessGroup

Adds a user group to this computer role.

Syntax

```
IAzRoleAssignment AddAccessGroup(DirectoryEntry group)
IAzRoleAssignment AddAccessGroup(SearchResult groupSr)
IAzRoleAssignment AddAccessGroup(string groupDn)
IAzRoleAssignment AddAccessGroup(IADsGroup grouplads)
```

Parameters

Specify one of the following parameters when using this method.

group	The directory entry for the group you want to add.
groupSr	The directory entry for a group specified as a search result.
groupDn	The group specified as a distinguished name.
grouplads	The IADs interface to the group.

Discussion

The AddAccessGroup(DirectoryEntry group) and AddAccessGroup(SearchResult group) methods are available only for .NET-based programs; call AddRoleAssignment for VBScript.

Return value

The computer role assignment that includes the new group.

Exceptions

AddAccessGroup may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid group or if it failed to create a role assignment because it cannot find the group.

AddRoleAssignment

Adds an empty user role assignment to the computer role.

Syntax

```
IRoleAssignment AddRoleAssignment()
```

Return value

Returns an empty role assignment. This role assignment is not stored in Active Directory until you call the RoleAssignment:Commit method.

Discussion

Any number of users can be assigned to a computer role and each of those users can have more than one role. Use this method to get an empty user role assignment for a computer role.

AddUser

Adds a user to this computer role.

Syntax

```
IAzRoleAssignment AddUser(DirectoryEntry user)
IAzRoleAssignment AddUser(SearchResult userSr)
IAzRoleAssignment AddUser(string userDn)
IAzRoleAssignment AddUser(IADsUser userIads)
```

Parameters

Specify one of the following parameters when using this method.

user	The directory entry for the user you want to add.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The computer role assignment that includes the new user.

Discussion

The AddUser(DirectoryEntry user) and AddAccessGroup(SearchResult user) methods are available only for .NET-based programs. Call AddRoleAssignment for VBScript.

Exceptions

AddUser may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid user or if it failed to create a role assignment because it cannot find the user.

ClearCustomAttributes

Clears the custom attributes for this computer role.

Syntax

```
void ClearCustomAttributes()
```

Commit

Commits any changes or updates to a computer role and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

This method does not validate changes. Call the Validate method before calling the `Commit` method.

Exceptions

`Commit` throws an `ApplicationException` if it could not find the computer role, could not find authorization data for the role, or failed to commit the computer role due to a communication error.

Delete

Marks the computer role for deletion from Active Directory.

Syntax

```
void Delete()
```

Exceptions

`Delete` throws an `ApplicationException` if it can't find the computer role, can't find authorization data for the zone, or failed to delete the role for another reason.

GetAccessGroup

Gets a user group assigned to this computer role given a specific role.

Syntax

```
IAzRoleAssignment GetAccessGroup(IRole role, DirectoryEntry group)
IAzRoleAssignment GetAccessGroup(IRole role, SearchResult groupSr)
IAzRoleAssignment GetAccessGroup(IRole role, string groupDn)
IAzRoleAssignment GetAccessGroup(IRole role, IADsGroup groupIAds)
```

Parameters

Specify the following parameter when using this method:

role	The role of the group.

Specify one of the following parameters when using this method.

group	The directory entry for the group.
groupSr	The directory entry for a group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The computer role assignment that includes the specified group (IAzRoleAssignment.TrusteeType==Group).

Discussion

Any number of user groups can be assigned to a computer role and each of those groups can have more than one role. Use this method to get the computer role assignment for a specific group and role.

The GetAccessGroup(IRole role, DirectoryEntry group) and GetAccessGroup(IRole role, SearchResult groupSr) methods are available only for .NET-based programs; call AddRoleAssignment for VBScript.

Exceptions

GetAccessGroup may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid group; or if it failed to get a role assignment because it cannot find the group.

GetAccessGroups

Gets the user groups assigned to this computer role.

Syntax

```
IRoleAssignments GetAccessGroups()
```

Return value

The collection of computer role assignments. Enumerate this object to get all of the `IRoleAssignment` objects for this computer role that represent groups (`IRoleAssignment.TrusteeType==Group`).

GetCustomAttributeContainer

Returns the directory entry for the parent container object for the custom attributes for this computer role.

Syntax

```
ICustomAttributeContainer GetCustomAttributeContainer()
```

Return value

The directory entry for the parent container object for the custom attributes for this computer role.

Discussion

The `GetCustomAttributeContainer` method is available only for .NET-based programs.

GetGroup

Returns the computer group associated with this computer role.

Syntax

```
DirectoryEntry GetGroup()
```

Return value

The directory entry for the computer group associated with the computer role.

Discussion

The GetGroup method is available only for .NET-based programs; call Group for VBScript.

GetRoleAssignment

Returns the user role assignment for a specified role and user.

Syntax

```
IRoleAssignment GetRoleAssignment(IRole role, string dn)
```

Parameters

Specify the following parameters when using this method.

role	The role for which you want a role assignment.
dn	The distinguished name of the user for which you want a role assignment.

Return value

The role assignment for the specified role and user.

Discussion

Any number of users can be assigned to a computer role and each of those users can have more than one role. Use this method to get the user role assignment for a specific user and role. To get the computer role assignment for a specific user, call the GetUser method.

Exceptions

`GetRoleAssignment` throws an `ArgumentNullException` if one of the specified parameter values is `null`.

GetRoleAssignmentById

Returns a role assignment, given a GUID.

Syntax

```
IRoleAssignment GetRoleAssignmentById(Guid id)
```

Parameter

Specify the following parameter when using this method:

id	The GUID of the role assignment.

Return value

The role assignment that has the specified GUID.

Discussion

This method returns a role assignment object given the GUID of the object.

Exceptions

`GetRoleAssignmentById` throws an `ArgumentNullException` if the specified parameter value is `null`.

GetRoleAssignments

Returns all the user role assignments under this computer role.

Syntax

```
IRoleAssignments GetRoleAssignments()
```

Return value

The collection of user role assignments for this computer role.

GetRoleAssignmentToAllADUsers

Returns the role assignment given to all Active Directory users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToAllADUsers(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignment.

Return value

The role assignment for the specified role.

Exceptions

GetRoleAssignmentToAllADUsers throws an ArgumentNullException if the specified parameter value is null.

GetRoleAssignmentToEveryone

Returns the role assignment given to all users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToEveryone(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignment.

Return value

The role assignment for the specified role.

Discussion

This method returns the role assignment for everyone with the specified role, including local users and groups.

Exceptions

`GetRoleAssignmentToEveryone` throws an `ArgumentNullException` if the specified parameter value is `null`.

GetUser

Gets a user assigned to this computer role.

Syntax

```
IAzRoleAssignment GetUser(IRole role, DirectoryEntry user)
IAzRoleAssignment GetUser(IRole role, SearchResult userSr)
IAzRoleAssignment GetUser(IRole role, string userDn)
IAzRoleAssignment GetUser(IRole role, IADsUser userIads)
```

Parameters

Specify the following parameter when using this method:

role	The role of the user.

Specify one of the following parameters when using this method.

user	The directory entry for the user you want to add.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The computer role assignment for the specified user and role.

Discussion

Any number of users can be assigned to a computer role and each of those users can have more than one role. Use this method to get the computer role assignment for a specific user and role. To get the user role assignment for a specific user, call the GetRoleAssignment method.

The GetUser(IRole role, DirectoryEntry user) and GetUser(IRole role, SearchResult userSr) methods are available only for .NET-based programs; call User for VBScript.

Exceptions

GetUser may throw the following exceptions:

- ApplicationException if cannot find the computer role in the zone; if it cannot find the specified role; if it cannot find authorization information for the zone; or if it failed to get the role assignment for some other reason.
- ArgumentNullException if a specified parameter value is null.

GetUsers

Gets the users assigned to this computer role.

Syntax

```
IRoleAssignments GetUsers()
```

Return value

The collection of computer roles. Enumerate this object to get all of the `IRoleAssignment` objects for this computer role that represent users (`IRoleAssignment.TrusteeType==User`).

SetCustomAttribute

Sets the custom attribute for this computer role.

Syntax

```
void SetCustomAttribute(string name, string value)
```

Parameters

Specify the following parameters when using this method:

name	The name of the custom attribute.
value	The value of the custom attribute

Return value

The collection of computer roles. Enumerate this object to get all of the IRoleAssignment objects for this computer role that represent users (IRoleAssignment.TrusteeType==User).

Validate

Validates the data in the ComputerRole object before any changes are committed to Active Directory. The method validates the following:

- The computer role name is not empty.
- The computer role name does not duplicate an existing computer role name in the zone.
- The computer role exists in the zone.
- An Active Directory computer group has been specified for the computer role.
- The specified computer group exists.

If the ComputerRole object is marked for deletion, the method skips validation tests.

Syntax

```
void Validate()
```

Exceptions

If the validation fails, Validate may throw an ApplicationException with a message indicating which test failed.

CustomAttributes

Syntax

```
string CustomAttributes {get; set;}
```

Property value

The custom attribute for this computer role.

Description

Gets or sets the description of this computer role.

Syntax

```
string Description {get; set;}
```

Property value

A string describing the computer role.

Group

Syntax

```
string Group {get; set;}
```

Property value

The Active Directory name of the computer group.

IsOrphan

Indicates whether the computer role is an orphan.

Syntax

```
bool IsOrphan {get;}
```

Property value

Returns `true` if this computer role cannot link to its Active Directory group.

Name

Gets or sets the name of the computer role.

Syntax

```
string Name {get; set;}
```

Property value

The name of the computer role.

Zone

Gets the zone of the computer role.

Syntax

```
IHierarchicalZone Zone {get;}
```

Property value

The zone in which the computer role is defined.

ComputerRoles

The ComputerRoles class manages a collection of computer roles.

Syntax

```
public interface IComputerRoles
```

Methods

The ComputerRoles class provides the following method:

GetEnumerator	Gets the enumerator you can use to enumerate all computer roles.

Properties

The ComputerRoles class provides the following property:

IsEmpty	Determines whether the collection is empty.

GetEnumerator

Returns an enumeration of ComputerRole objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all the ComputerRole objects.

IsEmpty

Indicates whether the collection of computer roles is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns `true` if there are no `ComputerRole` objects in the `ComputerRoles` object.

Computers

The Computers class manages a collection of Computer objects.

Syntax

```
public interface IComputers
```

Methods

The Computers class provides the following method:

GetEnumerator	Gets an enumerator you can use to enumerate all computer objects.

Properties

The ComputerRoles class provides the following property:

IsEmpty	Determines whether the collection is empty.

GetEnumerator

Returns an enumeration of Computer objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all the Computer objects.

IsEmpty

Determines whether the collection of computer objects is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns true if there are no Computer objects in the Computers object.

Delinea

ComputerUserUnixProfiles

The ComputerUserUnixProfiles class manages a collection of UserUnixProfile objects that represent computer users.

Syntax

```
public interface IComputerUserUnixProfiles : IUserUnixProfiles
```

Methods

The ComputerUserUnixProfiles class provides the following methods:

Find	Finds the user added to the specified computer.
GetEnumerator	Returns the enumeration of user profiles. (Inherited from UserUnixProfiles.)
Refresh	Reloads the cached user UNIX profiles. (Inherited from UserUnixProfiles.)

Properties

The ComputerUserUnixProfiles class provides the following properties:

Count	Gets the number of UserUnixProfile objects in this collection. (Inherited from UserUnixProfiles.)
IsEmpty	Determines whether the collection is empty. (Inherited from UserUnixProfiles.)

Find

Finds the UNIX user profile of the user of a specified computer.

Syntax

```
IHierarchicalUser Find(IHierarchicalZoneComputer computer)
```

Parameter

Specify the following parameter when using this method:

computer	The computer to search.

Return value

The UNIX user profile of the user of the specified computer; `null` if none exists. If there is more than one user, the first UNIX profile found is returned.

CustomAttribute

The CustomAttribute class contains a custom attribute. Available for .NET only.

Properties

The CustomAttribute class provides the following properties:

Name	The name of the custom attribute, specified as a string.
Value	The value of the custom attribute, specified as a string.

CustomAttributeContainer

The CustomAttributeContainer class contains a collection of custom attributes. Available for .NET only.

Methods

The CustomAttributeContainer class provides the following methods:

ClearCustomAttributes	Clears the custom attributes.
ICustomAttributes GetCustomAttributes	Interface to return the custom attributes.
SetCustomAttribute	Interface to set the custom attributes for this class.
ValidateCustomAttributes	Validates the custom attributes.

GetCustomAttributes

Gets the CustomAttributes.

Syntax

```
ICustomAttributes GetCustomAttributes(IHierarchicalZoneComputer computer)
```

Parameter

Specify the following parameter when using this method:

computer	The computer to search.

Return value

The custom attributes of the specified computer; `null` if none exists.

ValidateCustomAttributes

Validates the CustomAttributes.

Syntax

```
ValidateCustomAttributes(IHierarchicalZoneComputer computer)
```

Parameter

Specify the following parameter when using this method:

computer	The computer to search.

Return value

A boolean value; true if the custom attributes are valid. Otherwise, false.

CustomAttributes

The CustomAttributes class contains a set of custom attributes. Available for .NET only.

Methods

The CustomAttributeContainer class provides the following methods:

IEnumerator GetEnumerator	Interface to enumerate the custom attributes.

GetEnumerator

Returns an enumeration of CustomAttributes objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of CustomAttributes objects.

Entry

The Entry class contains methods and properties used to manage individual NIS map entries stored in Active Directory. This class is defined in the Centrify.DirectControl.NISMap.API namespace rather than the Centrify.DirectControl.API namespace.

Syntax

```
public class IEntry : ICloneable, IDisposable
```

Discussion

Each map entry consists of three primary fields: a key field, a value field, and an optional comment field.

The Entry class supports the methods and properties that apply to all .NET objects. In addition to those methods and properties, the Entry class provides some Delinea-specific methods and properties for managing the fields in NIS map records. Only the Delinea-specific methods and properties are described in this reference.

Methods

The Entry class provides the following Delinea-specific methods:

Clone	Makes a clone of the NIS map entry. Inherited from ICloneable.
Commit	Commits changes to the NIS map entry object and saves them in Active Directory.
Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
GetDirectoryEntry	Returns the DirectoryEntry attribute for the NIS map entry object.

Properties

The Entry class provides the following Delinea-specific properties:

Comment	Gets or sets the comment field associated with a specific key in a map entry.
IsReadable	Indicates whether the map entry is readable.
IsWritable	Indicates whether the map entry is writable.
Key	Gets or sets the key field in a map entry.
Map	Gets the NIS map associated with the map entry.
Value	Gets or sets the value field associated with a specific key in a map entry.

Commit

Commits the settings or changes for the map entry object to Active Directory.

Syntax

```
void Commit();
```

Exceptions

Commit throws an ApplicationException if it can't find the DirectoryEntry value or if the key or value is invalid.

Example

The following code sample illustrates using Commit to make changes to an existing NIS map entry to Active Directory:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Modify the value field for the “Workstation” map entry:
set entry = map.get("128.10.12.1")
entry.Value = "satellite1"
'Commit the changes to Active Directory
entry.Commit
wScript.Echo "NIS map entry “ & entry.Key & “: " & entry.Value
...
```

GetDirectoryEntry

Returns the DirectoryEntry object for the map entry object.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The directory entry for the map entry object.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Comment

Gets or sets the comment field for a specific NIS map entry.

Syntax

```
string Comment {get; set;}
```

Property value

The contents of the comment field for a specific NIS map entry.

Discussion

Each map entry consists of three primary fields: a key field, a value field, and an optional comment field. To use this property, you must be able to identify the map and the entry—the specific record in the map—for which you are setting or retrieving the comment.

Exceptions

`Comment` throws an `ArgumentException` if you try to set a value greater than 2048 characters.

Example

The following code sample illustrates using `Comment` to change the comment field in an existing NIS map entry:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Modify the Comment field for Workstation ”128.10.12.1” map entry:
set entry = map.get("128.10.12.1")
entry.Comment = "San Francisco, 5th floor, Accounting Dept."
'Commit the changes to Active Directory
entry.Commit
...
```

IsReadable

Indicates whether the map entry is readable for the user credentials presented to connect to Active Directory.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the map entry object is readable by the user, or false if the map entry object is not readable.

Discussion

This property returns a value of true if the user accessing the map entry object in Active Directory has sufficient permissions to read the entry properties.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=qa-slovenia,CN=unix,DC=quantum,DC=net")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Get the map entry specified
Set entry = map.get("128.10.12.1")
'Check whether the record is readable
If not entry.IsReadable then
    wScript.Echo “No read permission for this record”
end if
…
```

IsWritable

Indicates whether the map entry is writable for the user credentials presented to connect to Active Directory.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns true if the map entry object is writable by the user, or false if the map entry object is not writable.

Discussion

This property returns a value of true if the user accessing the map entry object in Active Directory has sufficient permissions to change the entry object's properties.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=qa-slovenia,CN=unix,DC=quantum,DC=net")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Get the map entry specified
Set entry = map.get("128.10.12.1")
'Check whether the record is writable
If not entry.IsWritable then
    wScript.Echo “No write permission for this record”
end if
…
```

Key

Gets or sets the key field for a NIS map entry.

Syntax

```
string Key {get; set;}
```

Property value

The contents of the key field for a NIS map entry.

Discussion

Each map entry consists of three primary fields: a key field, a value field, and an optional comment field.

Exceptions

`Key` throws an `ArgumentException` if you try to set a value that is null, empty, or greater than 1024 characters.

Example

The following code sample illustrates using `Key` to make changes to an existing NIS map entry and commit the changes to Active Directory.

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the NIS map named "generic map"
Set map = store.open("generic map")
'Modify the map entry fields for the “Key_Name” map record:
'entry.key = Key_Name
'entry.Value = Key_Value
'entry.comment = This is a sample generic map entry"
set entry = map.get("Key_Name")
entry.Key = "Modified_Key"
entry.Value = "Modified_Value"
entry.Comment = "Modified comment for the sample map entry"
'Commit the changes to Active Directory
entry.Commit
wScript.Echo "NIS map entry has been modified."
...
```

Map

Syntax

```
Map Map {get;}
```

Property value

The map containing this entry.

Value

Gets or sets the value field associated with a specific NIS map entry key.

Syntax

```
string Value {get; set;}
```

Property value

The value field associated with a specific NIS map entry key.

Discussion

Each map entry consists of three primary fields: a key field, a value field, and an optional comment field. The content and format of the value field depends on the type of NIS map you are working with. For example, if you are setting the value field in a generic map, the field can contain virtually any string that you want served for a corresponding key name.

If you are setting the value field in a netgroup map, the field lists the members of the group, separated by a blank space. Each member can be either a group name or a triple of the form (hostname,username,domainname). For example:

```
set map = store.open("netgrouop")
set entry = map.get("db_users")
entry.Value = "fin hr (,dean,ajax.org) (clone\*,,)"
```

If you are defining the value field in an auto.mount map entry, the field consists of the mount options, a tab character, and the network path to the file to consult for the mount point being defined. For example:

```
set map = store.open("auto.mount")
'Modify the value field of the “cdrom” mount point entry
set entry = map.get("cdrom")
entry.Value = "-fstype=nsfs,ro" & Chr(11) & ":/dev/sr0"
```

If you are defining the value field in an auto.master map entry, the field consists of the map file to consult, a tab character, and the mount options for the mount point being defined. For example:

```
set map = store.open("auto.mount")
'Modify the value field of the “/net” mount point entry
set entry = map.get("/net")
entry.Value = "-hosts" & Chr(11) & "-nosuid,nobrowse"
```

Exceptions

Value throws an ArgumentException if you try to set a value that is null, empty, or greater than 1024 characters.

Example

The following code sample illustrates using Value to set the value associated with a specified NIS map entry in a netgroup map:

```
…
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the NIS map named “netgroup”
set map = store.open("netgroup")
'Modify the value field of the “db_admins” entry
set entry = map.Get("db_admins")
entry.Value = "dbas dbowners (,dean,) (firebird,jon,)"
…
```

Group

Delinea uses existing Active Directory groups to manage the members of UNIX groups.

Syntax

```
public interface IGroup
```

Discussion

The Group class provides access to methods and properties that enable UNIX group profiles to be linked to Active Directory groups and that you can use to manage UNIX profiles associated with Active Directory groups. The additional UNIX-specific attributes that make up the UNIX profile for a group are stored and managed within the GroupUnixProfile object.

Methods

The Group class provides the following methods:

AddUnixProfile	Adds a new UNIX group profile to a zone.
Commit	Validates and saves changes to the Group object in Active Directory.
CommitWithoutCheck	Saves changes to the Group object in Active Directory without performing any validation.
GetDirectoryEntry	Returns the directory entry for an Active Directory group object from Active Directory.
GetRoleAssignmentsFromDomain	Returns the collection of all role assignments for a group in a specified domain.
GetRoleAssignmentsFromForest	Returns the collection of all role assignments for a group in a specified forest.
Refresh	Reloads the Group object data from the data in Active Directory.

Properties

The Group class provides the following properties:

AdsiInterface	Gets the IADs interface for an Active Directory group.
ADsPath	Gets the LDAP path for an Active Directory group.
ID	Gets the unique identifier for an Active Directory group.
UnixProfiles	Gets the GroupUnixProfiles object associated with an Active Directory group.

AddUnixProfile

Adds a new UNIX group profile to a zone.

Syntax

IGroupUnixProfile AddUnixProfile(IZone zone, int gid, string name)

IGroupUnixProfile AddUnixProfile(IZone zone, long gid, string name)

Parameters

Specify the following parameters when using this method.

zone	The individual zone to which you are adding a new UNIX group profile.
gid	The GID of the new UNIX group profile.
name	The name of the new UNIX group profile.

Return value

The UNIX group object created.

Discussion

The UNIX group profile includes the group name and the numeric group identifier (GID).

> **Note**: There are two versions of this method: one designed for COM-based programs that supports a 32-bit signed number for the gid argument and one designed for .NET-based programs that allows a 64-bit signed number for the gid argument.

Exceptions

AddUnixProfile may throw one of the following exceptions:

- ArgumentNullException if the zone parameter value is null.
- NotSupportedException if the specified zone has an unrecognized schema.

Example

The following code sample illustrates using AddUnixProfile in a script:

```
...
if (objGroup.UnixProfiles.Find(objZone) == null)
{
    long next_gid = 10000; // use 10000 as default gid
// Get the next available GID for this zone
if (objZone.NextAvailableGID \>= 0)
{
    next_gid = objZone.NextAvailableGID;
}
// Add this zone to the group
objGroupUnixProfile = objGroup.AddUnixProfile(objZone, next_gid, strUnixGroup);

// Save
objGroupUnixProfile.Commit();
...
```

Commit

Commits any changes or updates to the group object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

When you use this method, it checks and validates the data before saving it in Active Directory. Before saving, the method validates the following:

- The group name is a valid string that contains only letters (upper- or lowercase), numerals 0 through 9, and the hyphen (-) and underscore (_) characters.
- The GID value is a positive integer. Negative numbers are not allowed.
- The group name does not duplicate an existing group name.

Exceptions

`Commit` may throw one of the following exceptions:

- `ApplicationException` if any field in the UNIX group profile is invalid.
- `COMException` if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- `UnauthorizedAccessException` if you have insufficient permissions to commit the group object to Active Directory.

Example

The following code sample illustrates using `Commit` in a script:

```
...
if (objGroup.UnixProfiles.Find(objZone) == null)
{
    Console.WriteLine( strGroup + " was not a member of " + strZone);
    return;
}
else
    // Remove group
    objGroup.RemoveGroupUnixProfile(objZone);
    objGroup.Commit();
}
...
```

CommitWithoutCheck

Commits any changes or updates to the Group object and saves the changes to Active Directory without validating any of the data fields.

Syntax

void CommitWithoutCheck()

Discussion

Because this method does not perform any validation checking, it commits changes faster than the Group.Commit method.

Exceptions

CommitWithoutCheck may throw one of the following exceptions:

- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- UnauthorizedAccessException if you have insufficient permissions to commit the group object to Active Directory.

Example

The following code sample illustrates using CommitWithoutCheck in a script:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("ajax.org/UNIX/Zones/eur007")
'Identify the Active Directory group
set group = cims.GetGroupByPath("LDAP://CN=Subcontractors,
CN=EuropeanDiv,DC=ajax,DC=org")
'Set the UNIX profile associated with the group
group.SetGroupUnixProfile(zone, 8234, "subs")
'Update Active Directory without validation
group.CommitWithoutCheck
...
```

GetDirectoryEntry

Returns a DirectoryEntry object for the Active Directory group account from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry()
```

Return value

The directory entry for the UNIX group profile associated with the Active Directory group.

Discussion

The DirectoryEntry object represents the service connection point associated with the group in the zone.

> **Note**: This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

GetDirectoryEntry throws an ApplicationException if the directory object cannot be retrieved—for example, if it has not been committed.

Example

The following code sample illustrates using GetDirectoryEntry in a script:

```
...
//Identify the group you want to work with
IGroup group = cims.GetGroup("LDAP://CN=oracle1, CN=Users, DC=ajax, DC=org");
// Get the directory entry
DirectoryEntry groupEntry = group.GetDirectoryEntry();
// Rename the group
groupEntry.Rename("CN=oracle_dbas");
...
```

GetRoleAssignmentsFromDomain

Returns the collection of all role assignments explicitly assigned to a specified group—regardless of whether the role assignment is in a zone, computer-specific (computer override) zone, or computer role—within a specified domain.

Syntax

```
IRoleAssignments GetRoleAssignmentsFromDomain(string domain)
```

Parameters

Specify the following parameter when using this method:

domain	The domain to search for the group's role assignments.

Return value

A collection of role assignment objects representing all of the role assignments explicitly assigned to this group in the specified domain or in the currently joined domain.

Discussion

This method only returns role assignments explicitly assigned to the group. The method does not expand the group membership or return role assignments for groups nested under the specified group.

The method will look for stored credentials to access the specified domain. If there are no stored credentials, it uses the default credentials for the current user.

If you don't specify a domain by passing an empty string ("") to the method, the method returns role assignments from the currently joined domain.

Example

The following code sample illustrates using `GetRoleAssignmentsFromDomain` in a script:

```
...
// New Cims object
$cims = New-Object ("Centrify.DirectControl.API.Cims");
// Get IGroup object
$objGroupDn = "CN=group1,CN=Users,DC=domain,DC=com";
$objGroup = \$cims.GetGroup(\$objGroupDn);
// Get role assignments from domain
$objGroup.GetRoleAssignmentsFromDomain("domain.com")
...
```

GetRoleAssignmentsFromForest

Returns the collection of all role assignments explicitly assigned to a specified group—regardless of whether the role assignment is in a zone, computer-specific (computer override) zone, or computer role—within a specified forest.

Syntax

```
IRoleAssignments GetRoleAssignmentsFromForest(string forest)
```

Parameters

Specify the following parameter when using this method:

forest	The forest to search for the group's role assignments.

Return value

A collection of role assignments objects representing all of the role assignments explicitly assigned to this group in the specified forest or in the currently joined forest.

Discussion

This method only returns role assignments explicitly assigned to the group. The method does not expand the group membership or return role assignments for groups nested under the specified group.

The method will look for stored credentials to access the specified forest. If there are no stored credentials, it uses the default credentials for the current user.

If you don't specify a forest by passing an empty string ("") to the method, the method returns role assignments from the currently joined forest.

Example

The following code sample illustrates using `GetRoleAssignmentsFromForest` in a script:

```
...
// New Cims object
$cims = New-Object ("Centrify.DirectControl.API.Cims");
// Get IGroup object
$objGroupDn = "CN=group1,CN=Users,DC=domain,DC=com";
$objGroup = \$cims.GetGroup(\$objGroupDn);
// Get role assignments from forest
$objGroup.GetRoleAssignmentsFromForest("forest.com")
...
```

Refresh

Reloads the group object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the group information in the cached object to ensure it is synchronized with the latest information in Active Directory.

Example

The following code sample illustrates using Refresh in a script:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("ajax.org/UNIX/Zones/eur007")
'Identify the Active Directory group
set group = cims.GetGroupByPath("LDAP://CN=Subcontractors,
CN=EuropeanDiv,DC=ajax,DC=org")
'Modify the UNIX profile associated with the group
group.SetGroupUnixProfile(zone, 8234, "subcon07")
group.Commit
'Reload the group object from Active Directory
group.Refresh
wScript.Echo "Group Unix Profile Name: " & group.Name
...
```

AdsiInterface

Gets the IADsGroup interface for the group object from Active Directory.

Syntax

IADsGroup AdsiInterface {get;}

Property value

The IADsGroup interface for the group object.

Example

The following code sample illustrates using AdsiInterface in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://cn=eur007,cn=Zones,
cn=UNIX,dc=ajax,dc=org")
'Identify the Active Directory group
set objGroup =
cims.GetGroupByPath("LDAP://cn=Subcontractors,cn=EuropeanDiv,dc=ajax,dc=org")
'Get ADSI interface associated with the group
Set objAdsi = objGroup.AdsiInterface
...
```

ADsPath

Gets the LDAP path for the specified Active Directory group.

Syntax

```
string ADsPath {get;}
```

Property value

The LDAP path for the Active Directory group object.

Example

The following code sample illustrates using ADsPath for a group in a script:

```
…
'Identify the zone you want to work with
set objZone = cims.GetZone("sierra.com/program data/centrify/zones/market
research")
'Identify the Active Directory group
Set objGroup = cims.GetGroup("sierra.com/Groups/Managers")
‘Display the LDAP for the specified group
wScript.Echo "LDAP path: " & objGroup.ADsPath
…
```

ID

Gets the unique identifier for the specified Active Directory group.

Syntax

```
string ID {get;}
```

Property value

The GUID for the specified Active Directory group.

Example

The following code sample illustrates using ID for a group in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup = cims.GetGroupByPath("LDAP://CN=managers,
CN=Groups,DC=sierra,DC=com")
‘Display the ID for the specified group
wScript.Echo "Unique ID: " & objGroup.ID
...
```

UnixProfiles

Gets the GroupUnixProfiles object associated with a specified Active Directory group.

Syntax

IGroupUnixProfiles UnixProfiles {get;}

Property value

The GroupUnixProfiles object associated with the specified Active Directory group.

Discussion

The GroupUnixProfiles object contains information about the collection of group profiles associated with the Active Directory group in different zones.

Example

The following code sample illustrates using UnixProfiles in a script:

```
...
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Note: There is no cims.connect function.
// By default, this application will use the connection to the domain controller

// and existing credentials from the computer already logged in.
// Get the group object
IGroup objGroup = cims.GetGroupByPath(strGroup);
// Get the zone object
IZone objZone = cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN);
// Determine if the specified group is already a member of the zone.
// This method will either return a blank objGroupUnixProfile
// or one containing data
if (objGroup.UnixProfiles.Find(objZone) == null)
{
    Console.WriteLine( strGroup + " was not a member of " + strZone);
    return;
}
else
{
    // Remove group
    objGroup.RemoveGroupUnixProfile(objZone);
    objGroup.Commit();
}
...
```

Delinea

GroupInfo

The GroupInfo class contains methods and properties used to import and map UNIX group profiles to Active Directory groups. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

```
public interface IGroupInfo : IDisposable
```

Methods

The GroupInfo class provides the following methods:

Unexpected Link Text	Commits changes to the pending group object and saves them in Active Directory.
Unexpected Link Text	Marks the pending group profile for deletion from Active Directory.
Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
Unexpected Link Text	Returns the members of a pending import group.
Unexpected Link Text	Links the pending import group profile with the specified Active Directory group account.
Unexpected Link Text	Checks Active Directory for groups that match or conflict with a pending import group.

Properties

The GroupInfo class provides the following properties:

Unexpected Link Text	Gets the distinguished name (DN) of the import candidate.
Unexpected Link Text	Gets or sets the UNIX group identifier (GID) for the pending import group profile.
Unexpected Link Text	Gets the unique ID of the pending import group object.
Unexpected Link Text	Indicates whether the pending import group has been successfully imported.
Unexpected Link Text	Gets all of a pending import group's members.
Unexpected Link Text	Gets or sets the UNIX group name for a pending import group.
Unexpected Link Text	Gets the text string that describes the source of the pending import data.
Unexpected Link Text	Gets the status of the pending import group.
Unexpected Link Text	Gets a text string that provides detailed information about the status of the pending import group.

Unexpected Link Text	Gets the date and time that the pending group profiles were imported from the data source.

Commit

Commits any changes or updates to the pending group object and saves them in Active Directory.

Syntax

```
void Commit()
```

Delete

Marks the pending group profile object for deletion from Active Directory.

Syntax

```
void Delete()
```

Discussion

This method does not delete the pending group profile. After you mark the object for deletion, you must use the Unexpected Link Text method to commit changes to the object to Active Directory. When the Commit method is executed, the pending group profile is deleted from Active Directory to complete the operation.

Exceptions

Delete throws an UnauthorizedAccessException if you have insufficient access rights to remove the UNIX profile in the zone.

GetMembers

Returns the members of a pending import group.

Syntax

```
string GetMembers()
```

Return value

The collection of user profiles in the UserInfos object for the members of the pending import group.

Discussion

This method returns the collection of user profiles that are members of the pending import group.

Import

Imports the pending import group profile by associating the UNIX properties for the group with the specified Active Directory group account.

Syntax

```
void Import(IGroup group)
```

Parameter

Specify the following parameter when using this method:

group	The group for which you want to retrieve profile information.

Discussion

This method links the pending import group to an Active Directory account and removes the group from the pending import list.

UpdateStatus

Checks the Active Directory forest for matching or conflicting information that will allow or prevent a pending import group being imported.

Syntax

```
void UpdateStatus()
```

Discussion

This method searches Active Directory for a group name that matches the pending import group name and updates the pending import group properties with the results of the search. For example, if no Active Directory match is found or a UNIX profile already exists for the matching Active Directory group, the method updates the pending group's properties with that information.

> **Note:** Checking the Active Directory forest for potential matching candidates or conflicts can be a time-intensive operation. Therefore, you should consider the size and distribution of the forest and limit the number of pending import groups you are working with when using this method.

CandidateDN

Gets or sets the distinguished name (DN) of the import candidate.

Syntax

```
string CandidateDN {get; set;}
```

Property value

The matching Active Directory group object for the pending group profile, if one is found. If there's no matching candidate in Active Directory, null is returned.

Discussion

This property returns the Active Directory group account that appears to match the pending group profile. If there's an existing Active Directory group that matches the pending group, the pending import group can be mapped to that account. If no matching candidate is found in Active Directory, this property returns a null value.

GID

Gets or sets the UNIX group identifier (GID) for the pending import group profile.

Syntax

```
int GID {get; set;}

long GID {get; set;}
```

Property value

The UNIX group identifier (GID) for the pending group profile.

Discussion

There are two versions of this property: one designed for COM-based programs that supports a 32-bit signed number one designed for .NET-based programs that allows a 64-bit signed number. Therefore, the data type for the property can be an integer (int) or a long integer (long) depending on the programming language you use.

ID

Gets the unique ID of the pending import group object.

Syntax

```
string ID {get;}
```

Property value

The unique ID for the pending import group object.

IsImported

Determines whether the pending import group has been successfully imported.

Syntax

```
bool IsImported {get;}
```

Property value

Returns `true` if the pending import group has been imported, or `false` if the group has not been successfully imported.

Discussion

This property returns `true` if the pending import group has been imported, or `false` if the group has not been imported.

Members

Gets all of a pending import group's members.

Syntax

```
IGroupMembers Members {get;}
```

Property value

The user names for the members of a pending import group.

Name

Gets or sets the UNIX group name for a pending import group.

Syntax

```
string Name {get; set;}
```

Property value

The UNIX group name of a pending import group.

Source

Gets the text string that describes the source of the pending import data.

Syntax

```
string Source {get;}
```

Property value

A text string that describes the source of the pending import data.

Discussion

If the pending data was imported from a file, the property returns the source as File followed by the path to the file name imported. If the source of the pending import data was a NIS server, the property returns the NIS server name and domain. For example, if the source of the data was a file, the property returns a string similar to this:

```
File: C:\\Migration\\magnolia_groups
```

Status

Gets the status of the pending import group.

Syntax

```
StatusType Status {get;}
```

Property value

The status message for the pending import group.

Discussion

The status is determined by checking Active Directory for existing groups that match or conflict with the pending import group. The property returns a number that determines the icon displayed for the group in the console. The icons indicate whether a group is:

Ready to import	Info	
Has potential issues that should be resolved	Warning	
Cannot be imported	Error	

StatusDescription

Gets a text string that provides detailed information about the status of the pending import group.

Syntax

```
string StatusDescription {get;}
```

Property value

The status message for the pending import group.

Discussion

The status is determined by checking Active Directory for existing groups that match or conflict with the pending import group. The results are displayed in Access manager and in the Status tab of a pending group's Properties dialog box. The status description can also include details about the members of the group. For example, if checking the Active Directory forest revealed a group name or GID conflict with an existing group or another pending import group, the StatusDescription property might include information similar to this:

```
There is another pending imported group using the same GID.
There is another pending imported group using the same group name.
Group member:'alan' cannot be associated.
Group member:'rae' cannot be associated.
```

TimeStamp

Gets the date and time that the pending group profiles were imported from the data source.

Syntax

```
DateTime TimeStamp {get;}
```

Property value

The date and time that the pending group data was imported.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
Set objZone = cims.GetZone("w2k3.net/Acme/Zones/default")
‘Display the time groups where imported
Set objPendingGrps = objZone.GetImportPendingGroups
If not objPendingGrps is nothing then
wScript.Echo "Imported from source: ", objPendingGrps.TimeStamp
End if
…
```

Delinea

GroupInfos

The GroupInfos class contains methods and properties used to manage a collection of pending import group profiles. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

```
public interface IGroupInfos : IEnumerable\<IGroupInfo\>, IDisposable
```

Methods

The GroupInfos class provides the following methods:

Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
Unexpected Link Text	Returns the pending import group with the specified identifier from the collection of group profiles.
Unexpected Link Text	Returns an enumeration of GroupInfo objects.

Properties

The GroupInfos class provides the following properties:

Unexpected Link Text	Determines the total number of pending import group profiles defined in the collection represented by the GroupInfos object.
Unexpected Link Text	Determines whether the collection of pending import group profiles is empty.

Find

Returns the pending import group with the specified identifier from the collection of group profiles.

Syntax

```
IGroupInfo Find(string id)
```

Parameter

Specify the following parameter when using this method:

id	The unique identifier of the pending group profile for which you want to retrieve information.

Return value

The GroupInfo object for the specified pending import group.

GetEnumerator

Returns an enumeration of IGroupInfo objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of GroupInfo objects.

Count

Determines the total number of pending import group profiles defined in the GroupInfos collection.

Syntax

```
int Count {get;}
```

Property value

The number of pending import group profiles in the set.

Discussion

This property enumerates all of the profiles in the collection before it returns the Count value. If you only need to determine whether any import groups are pending, you should use the IsEmpty property for a faster response time.

IsEmpty

Determines whether the collection of pending import group profiles is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns `true` if there are no pending import group profiles in the `GroupInfos` object, or `false` if there is at least one pending import group profile in the object.

Discussion

Unlike the Count property, the `IsEmpty` property does not enumerate all of the pending import profiles in the collection before it returns a value. If you only need to determine whether any profiles are defined, you should call this property for a faster response.

GroupMember

The GroupMember class contains properties for working with the individual members of a pending import group. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

```
public interface IGroupMember : IDisposable
```

Methods

The GroupMember class provides the following method:

Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.

Properties

The GroupMember class provides the following properties:

CandidateDN	Gets or sets the distinguished name (DN) of the pending import group members.
Name	Gets or sets the UNIX user name for a pending import group member.

CandidateDN

Gets or sets the distinguished name (DN) of the pending import group members.

Syntax

```
string CandidateDN {get; set;}
```

Property value

The matching Active Directory group object for pending group profile, if one is found. If there's no matching candidate in Active Directory, nothing is returned.

Discussion

This property returns the Active Directory user account that appears to match the pending group member. If there's an existing Active Directory user that matches the pending import group member, the pending import group member can be mapped to that account.

Name

Gets or sets the UNIX user name for a pending import group member.

Syntax

```
string Name {get; set;}
```

Property value

The UNIX group name of a pending import group member.

GroupMembers

The GroupMembers class contains properties used to manage a collection of pending import group members. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

```
public interface IGroupMembers : IEnumerable\<IGroupMember\>, IDisposable
```

Methods

The GroupMembers class provides the following methods:

Add	Adds a new UNIX user as a member of the pending import group.
AddRange	Adds a list of new UNIX users as members of the pending import group.
Clear	Removes all of the group members from a pending import group.
Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
GetEnumerator	Returns an enumeration of GroupMember objects. Inherited from IEnumerable.
Remove	Removes the specified group member from the list of members in a pending import group.

Properties

The GroupMembers class provides the following properties:

Count	Determines the total number of group members in the pending import group.

Add

Adds a new UNIX user account as a member with the specified member name to the pending import group.

Syntax

IGroupMember Add(string memberName)

Parameter

Specify the following parameter when using this method:

memberName	The UNIX user name of the account you want to add to the pending import group.

Return value

The UNIX user you are adding as a member of the pending import group.

AddRange

Adds a list of new UNIX user profiles as members of the pending import group.

Syntax

```
void AddRange(ICollection string memberNames)
```

Parameter

Specify the following parameter when using this method:

memberNames	The list of UNIX user names you want to add as members of the pending import group.

Clear

Removes all of the group members from a pending import group.

Syntax

```
void Clear()
```

Discussion

This method enables you to import a pending import group profile without resolving membership conflicts or mapping group members to Active Directory users.

Remove

Removes the specified group member from the list of members in a pending import group.

Syntax

```
void Remove(IGroupMember member)
```

Parameter

Specify the following parameter when using this method:

member	The member you want to remove from the pending import group

Count

Determines the total number of group members in the pending import group.

Syntax

```
int Count {get;}
```

Property value

The number of group members in the pending import group.

Discussion

This property enumerates the list of members defined in the collection represented by the GroupMembers object.

Delinea

GroupUnixProfile

The GroupUnixProfile class manages the UNIX group profile information of an Active Directory group or a local group in a given zone.

Syntax

public interface IGroupUnixProfile

Discussion

An Active Directory or local group's zone-specific UNIX profile includes the numeric GID value and profile name directory.

Methods

The GroupUnixProfile class provides the following methods:

Commit	Commits changes to the GroupUnixProfile object to Active Directory.
Delete	Marks the UNIX group profile object for deletion from Active Directory.
GetDirectoryEntry	Returns the DirectoryEntry for a UNIX group profile from Active Directory.
Refresh	Reloads the GroupUnixProfile object data from the data in Active Directory.
Validate	Validates data in the GroupUnixProfile object before the changes are committed to Active Directory.

Properties

The GroupUnixProfile class provides the following properties:

ADsPath	Gets the LDAP path to the UNIX group profile.
Cims	Gets the Cims data for the group profile.
Group	Gets the Active Directory group to which the GroupUnixProfile object belongs (Active Directory groups only).
GroupID	Gets or sets the numeric group identifier (GID) for the group profile.
ID	Gets the unique identifier for the UNIX group profile.
IsForeign	Indicates whether the UNIX profile for a group is in a different forest than its corresponding Active Directory group (Active Directory groups only).
IsMembershipRequired	Determines whether an Active Directory group is a required group (Active Directory groups only).
IsOrphan	Indicates whether this UNIX group profile is an orphan (Active Directory groups only).
IsReadable	Indicates whether the Active Directory object is readable.
IsSFU	Indicates whether this UNIX group is an SFU zone profile (Active Directory groups only).
IsWritable	Indicates whether the Active Directory object is writable.

Members	Gets or sets the local group members of the UNIX group profile (local groups only).
Name	Gets or sets the group name of the UNIX group profile.
ProfileState	Gets or sets the profile state of the local group profile (local groups only).
Type	Gets the type of the UNIX group profile.
UnixEnabled	Determines whether the UNIX information is enabled.
Zone	Gets the zone object for the current GroupUnixProfile object.

Commit

Commits any changes or updates to the GroupUnixProfile object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

This method commits to Active Directory any new or changed values in the group UNIX profile. If an object is marked for deletion, calling this method completes the operation and deletes the object from Active Directory. The method also increments the next available group identifier (GID) by one, if applicable and permitted. The method does not validate the data before saving it in Active Directory.

Exceptions

Commit may throw one of the following exceptions:

- ApplicationException if it failed to get the directory entry of the group profile.
- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- ObjectAlreadyExistsException if the method receives a COMException with an LDAP object already exists error code.

Example

The following code sample illustrates using Commit in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://cn=onsite,cn=Zones,
cn=UNIX,dc=arcade,dc=com")
'Identify the Active Directory group
Set objGroup = cims.GetGroupByPath("CN=escalation,CN=support, DC=arcade,DC=com")

'Remove the membership requirement for the group
Set objGroup.IsMembershipRequired = false
'Save the changes in Active Directory
objGroup.commit
...
```

Delete

Marks the UNIX group profile object for deletion from Active Directory.

Syntax

```
void Delete()
```

Discussion

This method does not delete the group profile-- it only marks it for deletion. After you mark an object for deletion, you must call the Commit method to complete the operation.

Example

The following code sample illustrates using Delete in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://cn=eur007,cn=Zones,
cn=UNIX,dc=ajax,dc=org")
'Get the UNIX group profile you want to delete
set objProfile = objZone.GetGroupUnixProfileByGid(“905”)
objProfile.Delete
...
```

GetDirectoryEntry

Returns an instance of the directory entry for the group's UNIX profile from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The DirectoryEntry object for the UNIX group profile associated with the Active Directory group.

Discussion

The DirectoryEntry object represents the service connection point associated with the group in the zone.

> **Note:** This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Example

The following code sample illustrates using GetDirectoryEntry in a script:

```
...
// Identify the zone you want to work with
IZone zone = cims.GetZone("ajax.org/UNIX/Zones/NW_Support")
// Display the access control list
foreach (IGroupUnixProfile gpProfile in zone.GetGroupUnixProfiles))
{
// Get the directory entry
DirectoryEntry scp = gpProfile.GetDirectoryEntry();
Console WriteLine(scp.ObjectSecurity.GetSecurityDescriptorSddlForm
(AccessControlSections.Access));
}
...
```

Refresh

Reloads the GroupUnixProfile object data from the data in Active Directory.

Syntax

void Refresh()

Discussion

This method refreshes the group profile information in the cached object to ensure it is synchronized with the latest information in Active Directory.

Example

The following code sample illustrates using Refresh in a script:

```
…
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://cn=eur007,cn=Zones,
cn=UNIX,dc=ajax,dc=org")
'Get the UNIX group profile you want to work with
set objProfile = objZone.GetGroupUnixProfileByGid(“905”)
'Reload the group profile object from Active Directory
objProfile.Refresh
…
```

Validate

Validates the data in the GroupUnixProfile object before any changes are committed to Active Directory.

Syntax

```
void Validate()
```

Discussion

The method validates the following:

- The group name is a valid string that can contain only letters (upper- or lowercase), numerals 0 through 9, and the hyphen (-) and underscore (_) characters.
- The GID value is a positive integer. Negative numbers are not allowed.
- The group profile does not duplicate an existing group identifier (GID) or group name.

If the GroupUnixProfile object is marked for deletion, the method skips validation tests.

Exceptions

Validate throws an ApplicationException if any field in the UNIX group profile is invalid.

Example

The following code sample illustrates using Validate in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://cn=eur007,cn=Zones,
cn=UNIX,dc=ajax,dc=org")
'Get the UNIX group profile you want to work with
set objProfile = objZone.GetGroupUnixProfileByGid(“905”)
'Validate the UNIX profile associated with the group
objProfile.Validate
...
```

ADsPath

Gets the LDAP path to the UNIX group profile object.

Syntax

```
string ADsPath {get;}
```

Property value

The LDAP path to the UNIX group profile.

Example

The following code sample illustrates using `ADsPath in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup = cims.GetGroupByPath("LDAP://CN=managers,CN=groups,
DC=sierra,DC=com")
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
'Display the LDAP path for this group profile
wScript.Echo "LDAP Path: " & objGroupUnixProfile.ADsPath
...
```

Cims

Gets the Cims object for the group profile.

Syntax

```
Cims Cims {get;}
```

Property value

The Cims object for the group profile.

Discussion

This property serves as a shortcut for retrieving data.

Example

The following code sample illustrates using Cims in a script:

```
...
function doThings(gp2)
set objZone2 = gp2.Cims.GetZone("ajax.org/Zones/test")
gp2.Group.AddUnixProfile objZone2,objProfile.Gid,objProfile.Name
end function
set cims = CreateObject("Centrify.DirectControl.Cims3")
set objZone = cims.GetZone("ajax.org/Zones/default")
for each gp2 in objZone.GetGroupUnixProfiles
doThings gp2
next
...
```

Group

Gets the Active Directory group object associated with the specified GroupUnixProfile object.

Syntax

IGroup Group {get;}

Property value

The Active Directory group object associated with the UNIX group profile.

Example

The following code sample illustrates using Group in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/eur007")
'Get the UNIX group profile you want to work with
set objProfile = objZone.GetGroupUnixProfileByGid("905")
'Display the LDAP path to the profile's Active Directory group
wScript.Echo "LDAP path: " & objProfile.Group.ADsPath
...
```

GroupID

Gets the numeric group identifier (GID) for the group profile or sets a new GID for the specified Active Directory group in the specified zone.

Syntax

```
long GroupID {get; set;}
```

Property value

The numeric value of the UNIX GID for the UNIX profile in the zone.

Discussion

This property supports a 64-bit signed number for .NET modules.

Exceptions

`GroupID` throws and `InvalidOperationException` if the GID is `null` (that is, there is only a partial profile).

ID

Gets the unique identifier for the UNIX group profile from Active Directory.

Syntax

```
string ID {get;}
```

Property value

The unique identifier for this UNIX group profile.

Example

The following code sample illustrates using ID in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup = cims.GetGroupByPath("LDAP://CN=managers,CN=groups,
DC=sierra,DC=com")
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
'Display the unique ID for this group
wScript.Echo "Unique ID: " & objGroupUnixProfile.ID
...
```

IsForeign

Indicates whether the corresponding Active Directory group for a UNIX profile is in a different Active Directory forest than the forest associated with the group profile in the zone.

Syntax

```
bool IsForeign {get;}
```

Property value

Returns true if the UNIX profile is associated with an Active Directory group in a different forest.

Discussion

If the Active Directory group is in a different forest than the one associated with a top-level Delinea data object (Cims object), the property returns true.

> **Note:** This property is always false for newly-created groups before the group object is committed to Active Directory.

Example

The following code sample illustrates using IsForeign in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Check the forest for groups in the zone
For each profile in objZone.GetGroupUnixProfiles
if profile.IsForeign then
wScript.Echo profile.Name
end if
next
...
```

IsMembershipRequired

Determines whether the Active Directory group is a required group for its members.

Syntax

```
bool IsMembershipRequired {get; set;}
```

Property value

Returns true if the UNIX profile associated with an Active Directory group is marked as a required group.

Discussion

If this property is true, users cannot use the adsetgroups command to remove the group from the currently active set of groups. If this property is false, users who are members of the group can add or remove the group from their list of active groups at any time.

For more information about making a group required, see the *Administrator's Guide for Linux and UNIX*.

Exceptions

IsMembershipRequired throws an InvalidOperationException if there is only a partial profile.

Example

The following code sample illustrates using IsMembershipRequired in a script:

```
...
Set objZone = cims.GetZoneByPath("LDAP://CN=research,CN=zones,
CN=centrify,CN=program data,DC=sierra,DC=com")
'Get the UNIX group profile you want to work with
set objProfile = objZone.GetGroupUnixProfileByGid("905")
'Make this group a required group for its members
Set objProfile.IsMembershipRequired = true
...
```

IsOrphan

Indicates whether this UNIX group profile is an orphan.

Syntax

```
bool IsOrphan {get;}
```

Property value

Returns `true` if the GroupUnixProfile object has no corresponding Active Directory group object, or `false` if the object has a corresponding Active Directory group object.

Discussion

The UNIX group profile is an orphan if the corresponding Active Directory group object is missing.

Exceptions

`IsOrphan` throws an `ApplicationException` if the group profile does not exist.

Example

The following code sample illustrates using `IsOrphan in a script:

```
...
'Identify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/eur007")
'Check for orphan profiles
for each profile in objZone.GetGroupUnixProfiles
If profile.IsOrphan then
wScript.Echo profile.Name
end if
next
...
```

IsReadable

Indicates whether the group profile object in Active Directory is readable for the current user credentials.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the GroupUnixProfile object is readable, or false if the object is not readable.

Discussion

This property returns a value of true if the user accessing the group profile object in Active Directory has sufficient permissions to read its properties.

Example

The following code sample illustrates using IsReadable in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup =
cims.GetGroupByPath("LDAP://CN=testers,CN=groups,DC=sierra,DC=com")
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
'Check whether the object is readable
if not objGroupUnixProfile.IsReadable then
wScript.Echo "Denied read access. Exiting ...."
wScript.Quit
else
wScript.Echo "Read permission granted. Continuing ...."
wScript.Echo "Group Profile GID: " & objGroupUnixProfile.GID
end if
...
```

IsSFU

Indicates whether this group profile is an SFU zone profile.

Syntax

```
bool IsSFU {get;}
```

Property value

Returns `true` if the `GroupUnixProfile` object is an SFU zone profile.

Discussion

See Data storage for Delinea zones for a discussion of SFU zones.

IsWritable

Indicates whether the group profile object in Active Directory is writable for the current user's credentials.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns `true` if the `GroupUnixProfile` object is writable, or `false` if the object is not writable.

Discussion

This property returns a value of `true` if the user accessing the group profile object in Active Directory has sufficient permissions to change the group profile object's properties.

Example

The following code sample illustrates using `IsWritable` in a script:

```
...
set objZone =
cims.GetZoneByPath("LDAP://CN=research,CN=zones,CN=centrify,CN=program
data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup =
cims.GetGroupByPath("LDAP://CN=testers,CN=groups,DC=sierra,DC=com")
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
'Check whether the object is writable
if not objGroupUnixProfile.IsWritable then
wScript.Echo "Denied write access. Exiting ...."
wScript.Quit
else
wScript.Echo "Write permission granted. Continuing ...."
wScript.Echo "Group Profile GID: " & objGroupUnixProfile.GID
end if
...
```

Members

Gets an existing list of members or sets a new list of members for the UNIX group profile associated with the specified local group.

Syntax

```
string[] Members{get; set;}
```

Property value

The members of a local group.

Exceptions

Members throws an InvalidOperationException if the group you specify is not a local group.

Delinea

ProfileState

Gets the profile state of an existing local group or sets the profile of the specified local group.

Syntax

```
GroupProfileState ProfileState{get; set;}
```

Property value

The profile state of the specified local group.

Exceptions

ProfileState throws an InvalidOperationException if the group you specify is not a local group.

Name

Gets an existing name or sets a new name for the UNIX group profile associated with the specified Active Directory group in the specified zone.

Syntax

```
string Name {get; set;}
```

Property value

The UNIX group name for the UNIX profile in the zone.

Example

The following code sample illustrates using Name in a script:

```
…
set objZone =
cims.GetZoneByPath("LDAP://CN=research,CN=zones,CN=centrify,CN=program
data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup =
cims.GetGroupByPath("LDAP://CN=managers,CN=groups,DC=sierra,DC=com”)
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
wScript.Echo "Group Profile Name: " & objGroupUnixProfile.Name
…
```

`[title]: # (Type) [tags]: # (windows api) [priority]: # (21)

Type

Gets the type of the UNIX group profile.

Syntax

GroupUnixProfileType Type {get;}

Property value

Returns one of the following numeric values depending on how the UNIX group profile is stored:

- 0 indicates the group profile is a standard Delinea UNIX profile.
- 1 indicates the profile is a private group stored in a Private Groups container.
- 2 indicates the profile is a Delinea SFU profile.
- 3 indicates the profile is a local group.

Discussion

The Private group type is only applicable to early versions of Delinea software. It is not a valid group profile type in version 4.0 and later.

Example

The following code sample illustrates using Type in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data,DC=sierra,DC=com")
'Identify the Active Directory group
Set objGroup =
cims.GetGroupByPath("LDAP://CN=escalation,CN=support,DC=sierra,DC=com")
'Get the UNIX profile for the group in the zone
set objGroupUnixProfile = objGroupUnixProfiles.Find(objZone)
Select Case objGroupUnixProfile.Type
Case 0
wScript.Echo "Standard group"
Case 1
wScript.Echo "Private group"
Case 2
wScript.Echo "SFU group"
End Select
...
```

UnixEnabled

Determines whether the UNIX information is enabled.

Syntax

```
bool UnixEnabled {get; set;}
```

Property value

Set `true` if the UNIX information is enabled.

Example

For a code sample that uses the `UnixEnabled` property, see AddUnixProfile.

Zone

Gets the zone object for the current group unix profile.

Syntax

IZone Zone {get;}

Property value

The zone object for the UNIX group profile.

Discussion

This property serves as a shortcut for retrieving data.

GroupUnixProfiles

The GroupUnixProfiles class manages a collection of group profiles in a zone.

Syntax

```
public interface IGroupUnixProfiles
```

Discussion

The content of the collection of group profiles contained in the object depends on how the object was obtained:

- When you use GetUserUnixProfiles, the GroupUnixProfiles object returned enumerates all of the profiles defined for a specific Active Directory user across all zones in the current domain.
- When you use GetGroupUnixProfiles, the GroupUnixProfiles object returned enumerates all of the profiles defined for a specific Active Directory group in a specific zone.

Methods

The GroupUnixProfiles class provides the following methods:

GetEnumerator	Returns an enumeration of GroupUnixProfile objects.
Refresh	Reloads the GroupUnixProfiles object data from the data in Active Directory.

Properties

The GroupUnixProfiles class provides the following properties:

Count	Gets the total number of UNIX group profiles in the collection of GroupUnixProfiles for an Active Directory group.
IsEmpty	Indicates whether the GroupUnixProfiles object contains any UNIX group profiles.

GetEnumerator

Returns an enumeration of GroupUnixProfile objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

An enumeration of GroupUnixProfile objects.

Refresh

Reloads the GroupUnixProfiles object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the collection of group profiles in the cached object to ensure it is synchronized with the latest information in Active Directory.

Count

Gets the total number of UNIX group profiles defined in the GroupUnixProfiles collection for an Active Directory group or zone.

Syntax

```
int Count {get;}
```

Property value

The number of UNIX group profiles in the GroupUnixProfiles collection.

Discussion

This property enumerates all of the profiles in the collection before it returns the Count value. If you only need to determine whether any profiles are defined, use the IsEmpty property for a faster response time.

Example

The following code sample illustrates using Count in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data, DC=sierra, DC=com")
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data, DC=sierra, DC=com")
If objGroupUnixProfiles.IsEmpty then
wscript.echo "No profiles defined"
Else
wscript.echo objGroupUnixProfiles.Count & " profiles defined"
End if
...
```

IsEmpty

Indicates whether the collection of UNIX group profiles is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns true if there are no group profiles in the GroupUnixProfiles object, or false if there is at least one UNIX group profile in the object.

Discussion

Unlike the Count property, the IsEmpty property does not query all of the profiles in the collection before it returns a value. If you only need to determine whether any profiles are defined, call this property for a faster response.

Example

The following code sample illustrates using IsEmpty in a script:

```
...
set objZone = cims.GetZoneByPath("LDAP://CN=research,
CN=zones,CN=centrify,CN=program data, DC=sierra, DC=com")
If objGroupUnixProfiles.IsEmpty then
wscript.echo "No profiles defined”
Else
wscript.echo objGroupUnixProfiles.Count & " profiles defined”
End if
...
```

HierarchicalGroup

The HierarchicalGroup class manages the UNIX group profile information of an Active Directory group in a hierarchical zone, as well as local groups.

Syntax

```
public interface IHierarchicalGroup : IGroupUnixProfile
```

Methods

The HierarchicalGroup class provides the following methods:

Unexpected Link Text	Commits changes to the GroupUnixProfile object to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Marks the UNIX group profile object for deletion from Active Directory. (Inherited from Unexpected Link Text).
Unexpected Link Text	Returns the computer to which this group profile belongs.
Unexpected Link Text	Returns the directory entry for a UNIX group profile from Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Clears all property values so that all UNIX attributes for this user are inherited from the parent zone.
Unexpected Link Text	Reloads the GroupUnixProfile object data from the data in Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Resolves the effective profile.
Unexpected Link Text	Validates data in the GroupUnixProfile object before the changes are committed to Active Directory. (Inherited from Unexpected Link Text.)

Properties

The HierarchicalGroup class provides the following properties:

Unexpected Link Text	Gets the LDAP path to the UNIX group profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the Cims data for the group profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the effective GID of the group.
Unexpected Link Text	Indicates whether members of this group can remove the group from their currently active set of groups (not applicable to local groups).

Unexpected Link Text	Gets members of the local group (local groups only).
Unexpected Link Text	Gets the UNIX name of the group (not applicable to local groups).
Unexpected Link Text	Gets the profile state of the local group (local groups only).
Unexpected Link Text	Gets the Active Directory group to which the GroupUnixProfile object belongs (not applicable to local groups). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the numeric group identifier (GID) for the group profile. (Unexpected Link Text.)
Unexpected Link Text	Gets the unique identifier for the UNIX group profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether there is an effective GID for this group.
Unexpected Link Text	Indicates whether there is an effective membership requirement for this group (not applicable to local groups).
Unexpected Link Text	Indicates whether EffectiveMembers is defined for this group (local groups only).
Unexpected Link Text	Indicates whether there is an effective name for this group.
Unexpected Link Text	Indicates whether there is an effective profile state defined for this group (local groups only).
Unexpected Link Text	Determines whether the profile state is defined for this group (local groups only).
Unexpected Link Text	Indicates whether the UNIX profile for a group is in a different forest than its corresponding Active Directory group (not applicable to local groups). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the GID is defined for this group.
Unexpected Link Text	Determines whether Members is defined for this local group (local groups only).
Unexpected Link Text	Determines whether an Active Directory group is a required group (not applicable to local groups). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the membership requirement is defined for this group (not applicable to local groups).
Unexpected Link Text	Determines whether a name is defined for this group.
Unexpected Link Text	Indicates whether this UNIX group profile is an orphan (not applicable to local groups). (Inherited from Unexpected Link Text.)

Unexpected Link Text	Determines whether the Active Directory object is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether this UNIX group is an SFU zone profile (not applicable to local groups). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the Active Directory object is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets an existing list of members or sets a new list of members for the UNIX group profile associated with the specified local group (local groups only). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the group name of the UNIX group profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the profile state of a local group (local groups only). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the type of the UNIX group profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the UNIX information is enabled. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the zone associated with the UNIX group (inherited from Unexpected Link Text)
Unexpected Link Text	Gets the zone to which this group profile belongs.

GetComputer

Returns the computer to which this group profile belongs.

Syntax

```
IHierarchicalZoneComputer GetComputer()
```

Return value

Returns a hierarchical zone computer object specifying the computer to which this group profile belongs. Returns `null` if the group profile is not associated with a specific computer.

InheritFromParent

This method clears all current-level property values so that all property values are inherited from ancestor zones or from defaults.

Syntax

```
void InheritFromParent()
```

Discussion

This method clears all current-level property values so that all property values are inherited from ancestor zones or from defaults. This is a convenience method that is equivalent to resetting all properties to `null`.

ResolveEffectiveProfile

Resolves the effective profile for the group.

Syntax

```
void ResolveEffectiveProfile()
```

Discussion

This method resolves profiles of the group defined in the current zone, in parent zones, and in zone default values to determine the effective profile. If an error occurs, such as one of the parent zones not being accessible, the effective profile properties show the best-effort data retrieved before the error occurred.

EffectiveGid

Gets the GID of the group.

Syntax

```
long EffectiveGid {get;}
```

Property value

The GID in the UNIX group profile.

Exceptions

EffectiveGid throws an InvalidOperationException if the UNIX group profile does not include a GID.

EffectiveMembers

Gets the members of the local group.

Syntax

```
string[] EffectiveMembers {get;}
```

Property value

The members of the group.

Exceptions

EffectiveGid throws an InvalidOperationException if the UNIX group profile does not have members defined, or if this is not a local group profile and you attempt to set or get this property.

EffectiveIsMembershipRequired

Indicates whether members of this group can remove the group from their currently active set of groups.

Syntax

```
bool EffectiveIsMembershipRequired {get;}
```

Property value

Returns true if you can use the adsetgroups command to remove this group from your currently active set of groups.

Discussion

On most UNIX systems, a user can be a member of only a limited number of groups at one time. Because of this limitation, it is useful to be able to change a user's group membership by adding and removing groups when necessary.

You can use the adsetgroups command to manage the set of Active Directory groups that are available to a UNIX account. You also have the option to specify that membership in a specific group is required in a zone.

If you specify that a group is required, users who are members of the group cannot remove that group from their currently active set of groups. In that case, the EffectiveIsMembershipRequired property returns false.

Exceptions

IsEffectiveMembershipRequired throw an InvalidOperationException if there is only a partial profile.

EffectiveName

Gets the UNIX name of the group.

Syntax

```
string EffectiveName {get;}
```

Property value

The group name in the UNIX group profile.

EffectiveProfileState

Gets the profile state of the local group.

Syntax

```
GroupProfileState EffectiveProfileState {get;}
```

Property value

The profile state of the local group.

Exceptions

`EffectiveProfileState` throws an `InvalidOperationException` if the UNIX group profile does not have a profile state defined, or if this is not a local group profile and you attempt to get this property.

IsEffectiveGidDefined

Indicates whether there is an effective GID for this group.

Syntax

```
bool IsEffectiveGidDefined {get;}
```

Property value

Returns true if there is a UNIX group identifier (GID) in the UNIX group profile.

Discussion

If the Unexpected Link Text method has not been called, the value is resolved the first time this property is accessed.

IsEffectiveIsMembershipRequiredDefined

Indicates whether there is an effective membership requirement for this group.

Syntax

```
bool IsEffectiveIsMembershipRequiredDefined {get;}
```

Property value

Returns `true` if there is an effective membership requirement for this group.

Discussion

If the Unexpected Link Text method has not been called, the value is resolved the first time this property is accessed.

IsEffectiveMembersDefined

Indicates whether there is an effective members requirement for this local group.

Syntax

```
bool IsEffectiveMembersDefined {get;}
```

Property value

Returns `true` if there is an effective members requirement for this group.

Discussion

If the Unexpected Link Text method has not been called, the value is resolved the first time this property is accessed.

Exceptions

`IsEffectiveMembersDefined` throws an `InvalidOperationException` if this is not a local group profile and you attempt to get this property.

This property is only applicable to local groups.

IsEffectiveNameDefined

Indicates whether there is an effective name for this group.

Syntax

```
bool IsEffectiveNameDefined {get;}
```

Property value

Returns true if there is an effective name for this group.

Discussion

If the Unexpected Link Text method has not been called, the value is resolved the first time this property is accessed.

IsEffectiveProfileStateDefined

Indicates whether there is an effective profile state for this local group.

Syntax

```
bool IsEffectiveProfileStateDefined {get;}
```

Property value

Returns `true` if there is an effective profile state for this group.

Discussion

If the Unexpected Link Text method has not been called, the value is resolved the first time this property is accessed.

This property is only applicable to a local group profile.

Exceptions

`IsEffectiveProfileStateDefined` throws an `InvalidOperationException` if this is not a local group profile and you attempt to get this property.

IsGidDefined

Determines whether there is a GID defined for this group.

Syntax

```
bool IsGidDefined {get; set;}
```

Property value

Returns `true` if there is a GID defined for this group. Set this property false to clear the GID.

Exceptions

`IsGidDefined` throws an `InvalidOperationException` if the GID has not been defined and you attempt to set this property true.

IsMembersDefined

Indicates whether Members is defined for this local group.

Syntax

```
bool IsMembersDefined {get; set;}
```

Property value

Returns true if Unexpected Link Text is defined for this group. Set this property false to clear Unexpected Link Text.

Exceptions

IsMembers throws an InvalidOperationException if:

- Unexpected Link Text has not been defined and you attempt to set this property true.
- If this is not a local group and you attempt to set or get this property.

IsMembershipRequiredDefined

Determines whether the membership requirement is defined for this group.

Syntax

```
bool IsMembershipRequiredDefined {get; set;}
```

Property value

Returns true if the membership requirement is defined for this group. Set this property false to clear the Unexpected Link Text flag.

Exceptions

IsMembershipRequiredDefined throws an InvalidOperationException if the Unexpected Link Text flag has not been defined and you attempt to set this property true.

IsNameDefined

Determines whether the name is defined for this group.

Syntax

```
bool IsNameDefined {get; set;}
```

Property value

Returns `true` if the name is defined for this group. Set this property `false` to clear the name.

Exceptions

`IsNameDefined` throws an `InvalidOperationException` if the name has not been defined and you attempt to set this property true.

IsProfileStateDefined

Indicates whether there is a profile state defined for this local group.

Syntax

```
bool IsProfileStateDefined {get;}
```

Property value

Returns `true` if there is an a profile state defined for this group.

Discussion

Setting this property to `false` will clear Unexpected Link Text.

Exceptions

`IsProfileStateDefined` throws an `InvalidOperationException` if:

- Unexpected Link Text is not defined and you attempt to set this property to true.
- This is not a local group profile and you attempt to set or get this property.

Zone

Gets the zone to which this group profile belongs.

Syntax

```
IHierarchicalZone Zone {get;}
```

Property value

The zone to which this group profile belongs; `null` if this is a computer-specific profile.

Delinea

HierarchicalUser

The HierarchicalUser class manages the UNIX user profile information of an Active Directory user in a hierarchical zone.

Syntax

```
public interface IHierarchicalUser : IUserUnixProfile
```

Discussion

In hierarchical zones, both identity (profile data) and access (authorization data) are inherited, such that a user's effective identity or access are determined by all the profile data and all the access data at all levels of the hierarchy.

Profile data can be defined at any level: parent, child, or computer. It is possible to define a partial profile at any level — that is, leave one or more of the NSS fields blank. Although a complete profile is required to have access to a machine, a profile in a child zone can complete the missing fields from the parent zone. In the case of conflict, profile definitions in a child zone override the definition in the parent zone and computer-level definitions override all zone-level definitions.

On the other hand, role assignments do not override each other. Rather, they accumulate, such that a user's potential rights include all the rights granted by all the role assignments in the access tree. These are *potential* rights because rights granted to a user by a role assignment are effective only if the user has a complete profile defined for a zone.

In other words, when a computer joins a zone, the profile tree determines a pool of potential users, the access tree determines a different set of users with rights, and where the two intersect is the set of effective users.

See the Unexpected Link Text class for a user's Windows profile.

Methods

The HierarchicalUser class provides the following methods:

Unexpected Link Text	Returns a new user role assignment.
Unexpected Link Text	Commits changes to the userUnixProfile object to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Marks the UNIX user profile object for deletion from Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the computer to which this user profile belongs.
Unexpected Link Text	Returns the directory entry for a UNIX user profile from Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the effective user role assignments.
Unexpected Link Text	Returns the UNIX profile of the primary group of the user. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns a user role assignment for this UNIX user.
Unexpected Link Text	Returns all the user role assignments for this UNIX user.

Unexpected Link Text	Clears all property values so that all UNIX attributes for this user are inherited from the parent zone.
Unexpected Link Text	Reloads the userUnixProfile object data from the data in Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Resolves the effective profile to be used when the user logs on to the computer.
Unexpected Link Text	Resolves the effective roles for this user.
Unexpected Link Text	Validates data in the userUnixProfile object before the changes are committed to Active Directory. (Inherited from Unexpected Link Text.)

Properties

The HierarchicalUser class provides the following properties:

Unexpected Link Text	Gets the LDAP path to the UNIX user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the Cims data for the user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the contents of the effective GECOS field of the user profile.
Unexpected Link Text	Gets the hierarchical zone of the effective GECOS.
Unexpected Link Text	Gets the effective home directory of the user.
Unexpected Link Text	Gets the zone of the user's home directory.
Unexpected Link Text	Indicates whether this user uses an auto private group (not applicable to local user profiles).
Unexpected Link Text	Gets the user's effective logon name.
Unexpected Link Text	Gets the zone of the user's effective UNIX name.
Unexpected Link Text	Gets the effective primary group GID of the user.
Unexpected Link Text	Gets the zone of the primary group GID.

Unexpected Link Text	Gets the effective profile state of the local user (local user profiles only).
Unexpected Link Text	Gets the zone which defines the effective profile state
Unexpected Link Text	Gets the effective logon shell of the user.
Unexpected Link Text	Gets the zone of the effective logon shell.
Unexpected Link Text	Gets the effective UID of the user.
Unexpected Link Text	Gets the zone of the user's effective UID.
Unexpected Link Text	Gets or sets the contents of the GECOS field explicitly set in the user profile of the current zone.
Unexpected Link Text	Gets or sets the home directory of the user. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the unique identifier for the UNIX user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether there is an effective GECOS for this user.
Unexpected Link Text	Indicates whether there is an effective home directory defined for this user.
Unexpected Link Text	Indicates whether there is an effective name for this user.
Unexpected Link Text	Indicates whether a primary group is defined for this user.
Unexpected Link Text	Indicates whether there is an effective profile state for this local user (local user profiles only).
Unexpected Link Text	Indicates whether there is an effective shell defined for this user.
Unexpected Link Text	Indicates whether the user has an effective UID.
Unexpected Link Text	Indicates whether the auto private group flag is defined for this user (not applicable to local user profiles).
Unexpected Link Text	Indicates whether the UNIX profile for a user is in a different forest than its corresponding Active Directory user (not applicable to local user profiles). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the GECOS is defined in this profile.

Unexpected Link Text	Determines whether the home directory is defined in this profile.
Unexpected Link Text	Determines whether a name is defined in this profile.
Unexpected Link Text	Indicates whether this UNIX user profile is an orphan (not applicable to local user profiles). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether there is a GID defined for this user in this zone.
Unexpected Link Text	Gets or sets whether the profile state is defined in this local user profile (local user profiles only).
Unexpected Link Text	Determines whether the Active Directory object is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether this is a secondary profile (not applicable to local user profiles).
Unexpected Link Text	Indicates whether this user object uses the Microsoft Services for UNIX (SFU) schema extension (not applicable to local user profiles). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the shell is defined in this profile.
Unexpected Link Text	Determines whether the ID is defined in this profile.
Unexpected Link Text	Determines whether this user uses auto private groups (not applicable to local user profiles).
Unexpected Link Text	Determines whether the auto private group flag is defined (not applicable to local user profiles).
Unexpected Link Text	Determines whether the Active Directory object is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the user name of the UNIX user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the GID of the user's primary group. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the profile state of a local user profile (local user profiles only). (Inherited from Unexpected Link Text)
Unexpected Link Text	Gets or sets the user's default shell. (Inherited from UserUnUnexpected Link TextixProfile.)
Unexpected Link Text	Gets the type of the UNIX user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the UNIX information is enabled. (Inherited from Unexpected Link Text.)

Unexpected Link Text	Gets the user to whom this UNIX profile belongs (not applicable to local user profiles). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the user identifier (UID) for the user profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the zone associated with the UNIX user (inherited from Unexpected Link Text) Gets the zone to which this user profile belongs.

AddUserRoleAssignment

Adds a user role assignment to the user profile.

Syntax

IRoleAssignment AddUserRoleAssignment()

Return value

An empty user role assignment object. This role assignment is not stored in Active Directory until you call the RoleAssignment:[Commit](dev/windows-api/object-reference/computerrole/commit.md) method.

Discussion

This object is not saved to Active Directory until you set at least one property value and call the Unexpected Link Text method.

Example

The following code sample illustrates using AddUserRoleAssignment in a script:

```
...
IHierarchicalUser objUserUnixProfile = (IHierarchicalUser)
objZone.GetUserUnixProfile(objUser);
if (objUserUnixProfile == null)
{
    // New user for the zone
    objUserUnixProfile = objZone.AddUserPartialProfile(strUser);
}
IRole objRole = objZone.GetRole(strRole);
if (objRole == null)
{
    Console.WriteLine("Role " + strRole + " does not exist.");
    return;
}
IRoleAssignment asg = objUserUnixProfile.GetUserRoleAssignment(objRole);
if (asg != null)
{
    Console.WriteLine("Assignment already exist.");
    return;
}
else
{
    // assigning role to user
    asg = objUserUnixProfile.AddUserRoleAssignment();
    asg.Role = objZone.GetRole(strRole);
    asg.Commit();
    Console.WriteLine("Role " + strRole + " was successfully assigned to " + strUser
        \+ ".");
}
...
```

GetComputer

Returns the computer to which this user profile belongs.

Syntax

```
IHierarchicalZoneComputer GetComputer ()
```

Return value

Returns a hierarchical zone computer object specifying the computer to which this user profile belongs. Returns `null` if the user profile is not associated with a specific computer.

GetEffectiveUserRoleAssignments

Returns an enumeration of the effective user role assignments.

Syntax

```
IRoleAssignments GetEffectiveUserRoleAssignments()
```

Return value

An enumeration of the effective user role assignments for this user.

Discussion

The collection of effective role assignments is a combination of all the role assignments for this user in this zone and all parent zones. See the Unexpected Link Text class for a more complete discussion.

GetUserRoleAssignment

Returns the role assignment for a specific role for this user.

Syntax

```
IRoleAssignment GetUserRoleAssignment(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The role for which you want the role assignment.

Return value

The role assignment that associates this user with the specified role. Returns null if none exists.

Exceptions

GetUserRoleAssignment throws an ArgumentNullException if you pass null for the role parameter.

Example

The following code sample illustrates using GetUserRoleAssignment in a script:

```
...
// Create a CIMS object to interact with AD'
ICims cims = new Cims();
// Note: There is no cims.connect function.'
// By default, this application will use the connection to the domain controller

// and existing credentials from the computer already logged in.
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

IUser objUser = cims.GetUserByPath(strUser);
if (objUser == null)
{
    Console.WriteLine("User " + strUser + " does not exist.");
    return;
}
IHierarchicalUser objUnixUser = objZone.GetUserUnixProfile(objUser) as
IHierarchicalUser;
if (objUnixUser == null)
{
    objUnixUser = objZone.AddUserPartialProfile(strUser);
}
IRole objRole = objZone.GetRole(strRoleName);
if (objRole == null)
{
    Console.WriteLine("Role " + strRoleName + " does not exist.");
    return;
}
IRoleAssignment objAsg = objUnixUser.GetUserRoleAssignment(objRole);
if (objAsg == null)
{
    Console.WriteLine("Role assginment does not exist.");
    return;
}
else
{
    objAsg.Delete();
    Console.WriteLine("Role " + strRoleName + " was successfully removed from user "
        \+ strUser);
}
...
```

GetUserRoleAssignments

Returns an enumeration of the role assignments for this UNIX user.

Syntax

```
IRoleAssignments GetUserRoleAssignments()
```

Return value

An enumeration of the role assignments for this user in this zone.

Discussion

Call the Unexpected Link Text method to get the effective collection of role assignments, including those defined for this user in parent zones.

InheritFromParent

Clears all property values so that all UNIX attributes for this user are inherited from the parent zone.

Syntax

```
void InheritFromParent()
```

Discussion

This method clears all current-level property values so that all property values are inherited from ancestor zones or from defaults. This is a convenience method that is equivalent to resetting all properties to `null`.

ResolveEffectiveProfile

Resolves the profile for the user that is effective when the user logs on to the computer.

Syntax

```
void ResolveEffectiveProfile()
```

Discussion

This method resolves the profiles of the user in the current zone and parent zones, plus zone default values (if any), to determine effective profile values. If an error occurs, such as one of the parent zones not being accessible, the effective profile properties show the best-effort data retrieved before the error occurred.

You must call this method before calling any of the properties that return effective profile values.

ResolveEffectiveRoles

Resolves the effective roles for the user.

Syntax

```
void ResolveEffectiveRoles(IHierarchicalZone zone)
void ResolveEffectiveRoles(IHierarchicalZoneComputer computer)
```

Parameters

Specify one of the following parameters when using this method.

zone	The zone for which you want the group and user role assignments.
computer	The computer for which you want the group and user role assignments.

Discussion

This method resolves the groups and roles of the user in the specified zone or computer and parent zones. If you specify a zone, the method ignores computer-level roles and groups. If you specify a computer, the method considers only roles and groups defined for that computer. For a discussion of roles in hierarchical zones, see the Unexpected Link Text class.

EffectiveGecos

Gets the effective GECOS field of the user profile.

Syntax

```
string EffectiveGecos {get;}
```

Property value

The contents of the GECOS field of the effective profile for this user.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveGecosZone

Gets the zone in which the effective GECOS field is defined.

Syntax

```
IHierarchicalZone EffectiveGecosZone {get;}
```

Property value

The lowest-level hierarchical zone where the GECOS field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns null.

Call the Unexpected Link Text method to get or set the GECOS field for the current zone.

EffectiveHomeDirectory

Gets the effective home directory of the user.

Syntax

```
string EffectiveHomeDirectory {get;}
```

Property value

The contents of the home directory field of the effective profile for this user.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveHomeDirectoryZone

Gets the zone in which the effective home directory of the user is defined.

Syntax

```
IHierarchicalZone EffectiveHomeDirectoryZone {get;}
```

Property value

The lowest-level hierarchical zone where the home directory field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns `null`.

EffectiveIsUseAutoPrivateGroup

Indicates whether the effective user profile enables auto private groups.

Syntax

```
bool EffectiveIsUseAutoPrivateGroup {get;}
```

Property value

Returns `true` if the effective profile for this user enables auto private groups.

Discussion

Auto private group sets the user's UNIX profile name as the group name and the user's UID as the group GID.

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveName

Gets the user's effective logon name.

Syntax

```
string EffectiveName {get;}
```

Property value

The contents of the logon name field of the effective profile for this user.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveNameZone

Gets the zone in which the effective logon name of the user is defined.

Syntax

```
IHierarchicalZone EffectiveNameZone {get;}
```

Property value

The lowest-level hierarchical zone where the logon name field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns `null`.

EffectivePrimaryGroup

Gets the GID of the effective primary group from the user profile.

Syntax

```
long EffectivePrimaryGroup {get;}
```

Property value

The contents of the primary group field of the effective profile for this user.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveProfileState

Gets the effective profile state of the local user.

Syntax

```
UserProfileState EffectiveProfileState{get;}
```

Property value

The contents of the profile state filed of the effective profile for this local user.

Discussion

If `ResolveEffectiveProfile()` has not been called, this property will return either null, or the explicit value.

Exceptions

`EffectiveProfileState` throws an `InvalidOperationException` if this is not a local user profile and you attempt to get this property.

EffectiveProfileStateZone

Gets the zone which defines the effective profile state.

Syntax

```
IHierarchicalZone EffectiveProfileStateZone{get;}
```

Property value

The lowest-level hierarchical zone where the profile state field is defined. This value overrides any definitions in higher-level zones.

Discussion

If `ResolveEffectiveProfile()` has not been called, this property will always return null.

Exceptions

`EffectiveProfileStateZone` throws an `InvalidOperationException` if this is not a local user profile and you attempt to get this property.

Delinea

EffectivePrimaryGroupZone

Gets the zone in which the GID of the effective primary group of the user is defined.

Syntax

```
IHierarchicalZone EffectivePrimaryGroupZone {get;}
```

Property value

The lowest-level hierarchical zone where the primary group field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns `null`.

EffectiveShell

Gets the user's effective logon shell.

Syntax

```
string EffectiveShell {get;}
```

Property value

The contents of the logon shell field of the effective profile for this user.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveShellZone

Gets the zone in which the user's effective logon shell is defined.

Syntax

```
IHierarchicalZone EffectiveShellZone {get;}
```

Property value

The lowest-level hierarchical zone where the logon shell field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns `null`.

EffectiveUid

Gets the user's effective UID.

Syntax

```
long EffectiveUID {get;}
```

Property value

The contents of the UID field of the user's effective profile.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns the unresolved value at the current hierarchical level, or `null` if there is none.

EffectiveUidZone

Gets the zone in which the user's effective UID is defined.

Syntax

```
IHierarchicalZone EffectiveUIDZone {get;}
```

Property value

The lowest-level hierarchical zone where the UID field is defined. This value overrides any definitions in higher-level zones.

Discussion

You must call the Unexpected Link Text method before calling this property. If you don't do so, this property returns `null`.

Gecos

Gets or sets the GECOS field of the user profile in the current zone.

Syntax

```
string Gecos {get; set;}
```

Property value

The contents of the GECOS field.

Discussion

Call the Unexpected Link Text method to get the effective GECOS for this zone.

IsEffectiveGecosDefined

Indicates whether there is an effective value for the GECOS field for this user.

Syntax

```
bool IsEffectiveGecosDefined {get;}
```

Property value

Returns `true` if an effective value for the GECOS field exists for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectiveHomeDirectoryDefined

Indicates whether there is an effective home directory for this user.

Syntax

```
bool IsEffectiveHomeDirectoryDefined {get;}
```

Property value

Returns `true` if an effective home directory exists for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectiveNameDefined

Indicates whether there is an effective logon name for this user.

Syntax

```
bool IsEffectiveNameDefined {get;}
```

Property value

Returns `true` if an effective name exists for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectivePrimaryGroupDefined

Indicates whether there is an effective GID for this user.

Syntax

```
bool IsEffectivePrimaryGroupDefined {get;}
```

Property value

Returns `true` if an effective GID exists for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectiveProfileStateDefined

Indicates whether there is an effective profile state for this local user.

Syntax

```
bool IsEffectiveProfileStateDefined {get;}
```

Property value

Returns `true` if an effective profile state exists for this local user.

Exceptions

`IsEffectiveProfileStateDefined` throws an `InvalidOperationException` if this is not a local user profile and you attempt to get this property.

IsEffectiveShellDefined

Indicates whether there is an effective logon shell for this user.

Syntax

```
bool IsEffectiveShellDefined {get;}
```

Property value

Returns `true` if an effective logon shell exists for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectiveUidDefined

Indicates whether there is an effective UID for this user.

Syntax

```
bool IsEffectiveUidDefined {get;}
```

Property value

Returns true if there is an effective UNIX user identifier (UID) for this user.

Discussion

See the discussion of the Unexpected Link Text method.

IsEffectiveUseAutoPrivateGroupDefined

Indicates whether there is an effective auto private group flag setting for this user.

Syntax

```
bool IsEffectiveUseAutoPrivateGroupDefined {get;}
```

Property value

Returns `true` if there is an effective auto private group flag for this user.

Discussion

When auto private groups are enabled, the user's UNIX profile name is automatically used as the group name and the user's UID is used as the GID.

See the discussion of the `ResolveEffectiveProfile' method.

IsGecosDefined

Determines whether there is a GECOS field defined for this user in this zone.

Syntax

```
bool IsGecosDefined {get; set;}
```

Property value

Returns `true` if there is a GECOS field defined for this user. Set this property `false` to clear the GECOS field.

Exceptions

`IsGecosDefined` throws an `InvalidOperationException` if the GECOS field has not been defined and you attempt to set this property `true`.

IsHomeDirectoryDefined

Determines whether there is a home directory defined for this user in this zone.

Syntax

```
bool IsHomeDirectoryDefined {get; set;}
```

Property value

Returns `true` if there is a home directory defined for this user. Set this property `false` to clear the home directory.

Exceptions

`IsHomeDirectoryDefined` throws an `InvalidOperationException` if the home directory has not been defined and you attempt to set this property `true`.

IsNameDefined

Determines whether there is a logon name defined for this user in this zone.

Syntax

```
bool IsNameDefined {get; set;}
```

Property value

Returns `true` if there is a logon name defined for this user. Set this property `false` to clear the logon name.

Exceptions

`IsNameDefined` throws an `InvalidOperationException` if the name has not been defined and you attempt to set this property `true`.

IsProfileStateDefined

Determines whether the profile state is defined for this local user profile.

Syntax

```
bool IsProfileStateDefined {get; set;}
```

Property value

Returns `true` if there is a profile state defined for this user. Set this property `false` to clear the profile state.

Exceptions

`IsProfileStateDefined` throws an `InvalidOperationException` if:

- The profile state has not been defined and you attempt to set this property to `true`.
- This is not a local user profile and you attempt to set or get this property.

IsPrimaryGroupDefined

Determines whether there is a primary GID defined for this user in this zone.

Syntax

```
bool IsPrimaryGroupDefined {get; set;}
```

Property value

Returns `true` if there is a primary GID defined for this user. Set this property `false` to clear the GID.

Discussion

The user's primary group identifier (GID) can be associated with an Active Directory group or be a separate "dedicated-user" group that is only used in the UNIX operating environment. This property indicates whether a group profile for that GID has been defined in the zone.

Exceptions

`IsPrimaryGroupDefined` throws an `InvalidOperationException` if the primary GID has not been defined and you attempt to set this property `true`.

IsSecondary

Indicates whether the profile in this zone is a secondary profile.

Syntax

```
bool IsSecondary {get;}
```

Property value

Returns `true` if this is a secondary profile. Returns `false` if this is a primary profile.

IsShellDefined

Determines whether there is a logon shell defined for this user in this zone.

Syntax

```
bool IsShellDefined {get; set;}
```

Property value

Returns `true` if there is a logon shell defined for this user. Set this property `false` to clear the shell.

Exceptions

`IsShellDefined` throws an `InvalidOperationException` if the logon shell has not been defined and you attempt to set this property `true`.

IsUidDefined

Determines whether there is a UID defined for this user in this zone.

Syntax

```
bool IsUidDefined {get; set;}
```

Property value

Returns `true` if there is a UID defined for this user. Set this property `false` to clear the UID.

Exceptions

`IsUidDefined` throws an `InvalidOperationException` if the UID has not been defined and you attempt to set this property `true`.

IsUseAutoPrivateGroup

Determines whether the user uses auto private groups.

Syntax

```
bool IsUseAutoPrivateGroup {get; set;}
```

Property value

Returns `true` if this user uses an auto private group.

Discussion

When auto private groups are enabled, the user's UNIX profile name is automatically used as the group name and the user's UID is used as the GID.

IsUseAutoPrivateGroupDefined

Determines whether the auto private group flag is defined for this user in this zone.

Syntax

```
bool IsUseAutoPrivateGroupDefined {get; set;}
```

Property value

Returns `true` if the auto private group flag is defined for this user. Set this property `false` to remove the flag definition from the profile.

Discussion

When auto private groups are enabled, the user's UNIX profile name is automatically used as the group name and the user's UID is used as the GID.

Exceptions

`IsUseAutoPrivateGroupDefined` throws an `InvalidOperationException` if the auto private group flag has not been defined and you attempt to set this property `true`.

Zone

Gets the zone to which this user profile belongs.

Syntax

```
IHierarchicalZone Zone {get;}
```

Property value

The zone to which this user profile belongs; `null` if this is a computer-specific profile.

Delinea

HierarchicalZone

The HierarchicalZone class represents a hierarchical zone.

Syntax

```
public interface IHierarchicalZone : IZone
```

Discussion

The HierarchicalZone class inherits many methods and properties from the Zone class, but adds support for partial profiles and inheritable roles. Under hierarchical zones, both identity (profile data) and access (authorization data) are inherited, such that a user's effective identity or access are determined by all the profile data and all the access data at all levels of the hierarchy.

See Unexpected Link Text for a discussion of profile and access inheritance.

Methods

The HierarchicalZone class provides the following methods:

Unexpected Link Text	Adds an empty role assignment to a group
Unexpected Link Text	Creates a computer role under this zone.
Unexpected Link Text	Adds a partial profile for a specified group.
Unexpected Link Text	Adds a partial profile for a specified local group.
Unexpected Link Text	Adds a partial profile for a specified local user.
Unexpected Link Text	Adds an MIT Kerberos realm trusted user to this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Adds an empty role assignment.
Unexpected Link Text	Adds a partial profile for a specified user.
Unexpected Link Text	Commits changes to the group object to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Creates a command right for the zone.
Unexpected Link Text	Creates a pending imported group in this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Creates a pending imported user in this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Creates a network application access right.
Unexpected Link Text	Creates a PAM application access right.
Unexpected Link Text	Creates a role in the zone.
Unexpected Link Text	Creates an SSH application access right.
Unexpected Link Text	Creates a Windows application access right.
Unexpected Link Text	Creates a Windows Desktop access right.

Unexpected Link Text	Marks the zone for deletion from Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Generates predefined SSH and PAM rights in this zone.
Unexpected Link Text	Generates predefined user roles in this zone.
Unexpected Link Text	Returns a group assigned to this zone given a role for the group.
Unexpected Link Text	Returns an enumeration of groups in the zone.
Unexpected Link Text	Returns an enumeration of this zone's child zones.
Unexpected Link Text	Returns the privileged command right with a specific name or GUID.
Unexpected Link Text	Returns an enumeration of all the privileged command rights in the zone.
Unexpected Link Text	Returns the computer profile in the zone given the distinguished name of the profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns a specific computer role under this zone.
Unexpected Link Text	Returns an enumeration of all the computer roles under this zone.
Unexpected Link Text	Returns an enumeration of all the computers in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the Active Directory object for the Computers node. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the Active Directory object for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the display name of this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns all the command rights that can be assigned to users in the zone, including inherited rights.
Unexpected Link Text	Returns all the network access rights that can be assigned to users in the zone, including inherited rights.
Unexpected Link Text	Returns all the PAM application access rights that can be assigned to users in the zone, including inherited rights.
Unexpected Link Text	Returns all the user roles that can be assigned to users in the zone, including inherited roles.
Unexpected Link Text	Returns all the SSH application access rights that can be assigned to users in the zone, including inherited rights.
Unexpected Link Text	Returns an enumeration of effective users under this zone.
Unexpected Link Text	Returns all the Windows application access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.
Unexpected Link Text	Returns all the Windows desktop access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.
Unexpected Link Text	Returns all the Windows users in the zone, including users inherited from zones higher in the hierarchy.
Unexpected Link Text	Returns the DirectoryEntry of the local groups container. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the local UNIX group profile for a specified group name in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns a local group profile using the distinguished name (DN) of the profile. (Inherited from Unexpected Link Text.)

Unexpected Link Text (Int32)	Returns the local group profile using the Group Identifier (GID). This method is exposed to the .COM interface. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns a list of the local group profiles in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the directory entry of the local users container. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the local user profile using the specified user name. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the local user profile specified by the distinguished name (DN) of the profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text (Int32)	Returns the local user profile using the User Identifier (UID). This method is exposed to the .COM interface (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns a list of the local user profiles in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the specified network access right.
Unexpected Link Text	Returns all the network access rights that can be assigned to users in the zone.
Unexpected Link Text	Returns the Active Directory object for the Groups container. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the UNIX group profile in this zone for the specified Active Directory group. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the UNIX group profile in this zone for the Active Directory group specified by distinguished name. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the UNIX group profile in this zone for the Active Directory group specified by group name. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns an enumeration of the UNIX groups in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the group with the specified ID pending import. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns an enumeration of groups pending import to this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the user with the specified ID pending import. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns an enumeration of users pending import to this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	VBScript interface to access NSS variables.
Unexpected Link Text	VBScript interface to obtain all NSS variable names.
Unexpected Link Text	Returns the PAM application access right with the specified name.
Unexpected Link Text	Returns an enumeration of all the PAM application rights in the zone.
Unexpected Link Text	Returns the primary profile for the specified user.
Unexpected Link Text	Returns the role with the specified name or GUID.
Unexpected Link Text	Returns the role assignment for the specified role and trustee.
Unexpected Link Text	Returns the role assignment for the specified GUID.

Unexpected Link Text	Returns an enumeration of all the role assignments in the zone.
Unexpected Link Text	Returns the role assignment given to all Active Directory users who have a specified role.
Unexpected Link Text	Returns the role assignment given to all UNIX users who have a specified role.
Unexpected Link Text	Returns an enumeration of all the roles in the zone.
Unexpected Link Text	Returns an enumeration of the secondary profiles for the specified user.
Unexpected Link Text	Returns the SSH application access right with the specified name.
Unexpected Link Text	Returns an enumeration of all the SSH application rights in the zone.
Unexpected Link Text	Returns all role assignments under this zone, including role assignments for computer roles and computers.
Unexpected Link Text	Returns an enumeration of all the user profiles for the specified user.
Unexpected Link Text	Returns an enumeration of all the user role assignments in the zone.
Unexpected Link Text	Returns the specified Windows application right.
Unexpected Link Text	Returns all the Windows application rights in the zone.
Unexpected Link Text	Returns all the Windows computers in the zone.
Unexpected Link Text	Returns the specified Windows desktop right.
Unexpected Link Text	Returns all the Windows desktop rights in the zone.
Unexpected Link Text	Returns the directory entry of the Users container. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the UNIX user profile in this zone for the user specified by distinguished name. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns the UNIX user profile in this zone for the user specified by user name. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns an enumeration of all the UNIX user profiles in the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the group has a profile in this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether a UNIX profile exists in the zone for the specified local group. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether a UNIX profile exists in the zone for the specified local user. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Adds a computer zone to a computer object in this zone.
Unexpected Link Text	Refreshes the data in this object instance from the data stored in Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	VBScript interface to set the values of NSS variables.
Unexpected Link Text	Indicates whether the specified user has a profile in this zone. (Inherited from Unexpected Link Text.)

Properties

The HierarchicalZone class provides the following properties:

Unexpected Link Text	Gets the IADs interface of the zone object in Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the LDAP path to the zone object. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the attribute used to store the password hash for an agentless client. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets an enumeration of available user login shells. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the Cims object managing this zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the default group for new users. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the default login directory for new users. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the default login shell for new users. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the zone to use for default zone values. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the description of the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the full name of the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether auto-provisioning of group profiles is enabled for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the default group name.
Unexpected Link Text	Gets the unique identifier for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether this is a child zone.
Unexpected Link Text	Indicates whether the group default name is defined.
Unexpected Link Text	Indicates whether this is a hierarchical zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets whether Next GID value is configured for this zone.
Unexpected Link Text	Gets or sets whether Next UID value is configured for this zone.

Unexpected Link Text	Indicates whether this zone object in Active Directory is readable with the current user credentials. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the zone uses the Microsoft Services for UNIX (SFU) schema extension. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether this is a TruncateName zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the UseAutoPrivateGroup flag is defined.
Unexpected Link Text	Determines whether the user default GECOS is defined in this profile.
Unexpected Link Text	Determines whether the user default home directory is defined in this profile.
Unexpected Link Text	Determines whether the user default name is defined in this profile.
Unexpected Link Text	Determines whether the user default primary group is defined in this profile.
Unexpected Link Text	Determines whether the user default role is defined in this profile.
Unexpected Link Text	Determines whether the user default login shell is defined in this profile.
Unexpected Link Text	Indicates whether this zone object is writable using the provided credential. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the license container for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the master domain controller for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether Active Directory group membership must be maintained. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the next GID to be used when adding a group (32-bit for COM programs). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the next UID to be used when adding a user (32-bit for COM programs). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the next GID to be used when adding a group (64-bit for .NET modules). (Inherited from Unexpected Link Text.)

Unexpected Link Text	Gets or sets the next UID to be used when adding a user (64-bit for .NET modules). (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the NIS domain associated with this SFU zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the map of profile variables.
Unexpected Link Text	Gets or sets the parent of this zone.
Unexpected Link Text	Gets or sets the list of GIDs not to be used when adding groups. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the list of UIDs not to be used when adding users. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the schema of the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the Active Directory domain associated with this SFU zone for retrieving SFU information. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether to use the Apple algorithm to automatically generate the GID when adding a group. The Apple algorithm is based on the globally unique identifier (GUID) for the object.
Unexpected Link Text	Determines whether to use the Apple algorithm to automatically generate the UID when adding a user. The Apple algorithm is based on the globally unique identifier (GUID) for the object.
Unexpected Link Text	Determines whether to use the Delinea algorithm to automatically generate the GID when adding a group. The Delinea algorithm is based on the security identifier (SID) for the object.
Unexpected Link Text	Determines whether this zone defaults to use an auto private group when adding a zone user.
Unexpected Link Text	Determines whether to use the Delinea algorithm to automatically generate the UID when adding a user. The Delinea algorithm is based on the security identifier (SID) for the object.
Unexpected Link Text	Determines whether to use the NextGID property when adding a group.
Unexpected Link Text	Determines whether to use the NextUID property when adding a user.
Unexpected Link Text	Indicates whether auto-provisioning of user profiles is enabled for the zone. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the default GECOS field for new user profiles.
Unexpected Link Text	Gets or sets the user default GID when adding a new user profile.
Unexpected Link Text	Gets or sets the default user name for a new user profile.

Unexpected Link Text	Gets or sets the user default GID for new user profiles; for use in VBScript scripts.
Unexpected Link Text	Gets or sets the default role for a new user profile.
Unexpected Link Text	Gets the version number of the data schema. (Inherited from Unexpected Link Text.)

AddAccessGroup

Adds an empty role assignment to a group.

Syntax

```
IHzRoleAssignment AddAccessGroup(DirectoryEntry groupDE)
IHzRoleAssignment AddAccessGroup(SearchResult groupSR)
IHzRoleAssignment AddAccessGroup(string groupDn)
IHzRoleAssignment AddAccessGroup(IAdsGroup groupIAds)
```

Parameters

Specify one of the following parameters when using this method.

groupDE	The directory entry for the group.
groupSr	The directory entry for the group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The computer role assignment.

Discussion

The role assignment is not stored in Active Directory until you call the Unexpected Link Text method.

The AddAccessGroup(DirectoryEntry groupDE) and AddAccessGroup(SearchResult groupSR) methods are available only for .NET-based programs; call Unexpected Link Text for VBScript.

Exceptions

AddAccessGroup may throw one of the following exceptions:

- ApplicationException if the specified parameter is not a group or the method cannot find the group.
- ArgumentNullException if you pass a null parameter.

Example

The following code sample illustrates using the AddAccessGroup and GetAccessGroup methods in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
   return;
}
IRole role = objZone.GetRole(strRole);
if (role == null)
{
```

```
    Console.WriteLine(strRole + " does not exist in zone.");
}
else if (objZone.GetAccessGroup(role, strGroup) != null)
{
    Console.WriteLine("Role assignment already exist.");
}
else
{
    // assign a role to the group
    IRoleAssignment zag = objZone.AddAccessGroup(strGroup);
    zag.Role = role;
    zag.Commit();
}
…
```

AddComputerRole

Creates a computer role under this zone.

Syntax

```
IComputerRole AddComputerRole(string name)
```

Parameters

Specify the following parameter when using this method:

name	The name of the computer role you want to add.

Return value

AddGroupPartialProfile

Adds a partial profile for the specified group to the zone.

Syntax

IHierarchicalGroup AddGroupPartialProfile(DirectoryEntry groupDE)

IHierarchicalGroup AddGroupPartialProfile(SearchResult groupSR)

IHierarchicalGroup AddGroupPartialProfile(string groupDn)

IHierarchicalGroup AddGroupPartialProfile(IAdsGroup groupIAds)

Parameters

Specify one of the following parameters when using this method.

groupDE	The directory entry for the group for which you want a partial profile.
groupSr	The directory entry for a group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The hierarchical group object that represents the group profile.

Discussion

This method creates a new group profile with values set for the Cims and Group properties. If the zone is an SFU zone, then this method also sets a value for the NISDomain property. You can then add other properties to the profile.

The profile is not stored in Active Directory until you call the Unexpected Link Text method.

The AddAccessGroup(DirectoryEntry groupDE) and AddAccessGroup(SearchResult groupSR) methods are available only for .NET-based programs; call Unexpected Link Text for VBScript.

Exceptions

If you pass a null parameter, AddGroupPartialProfile throws the exception ArgumentNullException.

Example

The following code sample illustrates using the AddGroupPartialProfile method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone = cims.GetZoneByPath("cn=" + strZone + "," +
strContainerDN) as
IHierarchicalZone;
// Load the unix profiles associated with the group
// Determine if the specified group is already a member of the zone.
// This method will either return a blank objGroupUnixProfile
// or one containing data
IGroupUnixProfile objGroupUnixProfile;
if (objZone.GetGroupUnixProfileByName(strUnixGroup) == null)
{
    // Add this zone to the group
    objGroupUnixProfile = objZone.AddGroupPartialProfile(strGroup);
    objGroupUnixProfile.Name = strUnixGroup;
    // Save
```

```
    objGroupUnixProfile.Commit();
}
...
```

AddRoleAssignment

Adds an empty role assignment to the zone.

Syntax

```
IRoleAssignment AddRoleAssignment()
```

Return value

An empty role assignment object. This role assignment is not stored in Active Directory until you call the RoleAssignment:[Commit](dev/windows-api/object-reference/computerrole/commit.md) method.

AddLocalGroupPartialProfile

Adds a partial profile for the specified group to the zone.

Syntax

```
IHierarchicalUser AddlocalGroupPartialProfile(string groupName)
```

Parameters

Specify groupName; the name of the local group.

Return value

The hierarchical group object that represents the local group profile.

Exceptions

If you pass a null parameter, AddLocalGroupPartialProfile throws the exception ArgumentNullException.

AddLocalUserPartialProfile

Adds a partial profile for the specified user to the zone.

Syntax

```
IHierarchicalUser AddLocalUserPartialProfile(string userName)
```

Parameters

Specify `userName`; the user name of the local user.

Return value

The hierarchical user object that represents the local user profile.

Exceptions

If you pass a `null` parameter, `AddLocalUserPartialProfile` throws the exception `ArgumentNullException`.

AddUserPartialProfile

Adds a partial profile for the specified user to the zone.

Syntax

IHierarchicalUser AddUserPartialProfile(DirectoryEntry userDE)

IHierarchicalUser AddUserPartialProfile(SearchResult userSR)

IHierarchicalUser AddUserPartialProfile(string userDn)

IHierarchicalUser AddUserPartialProfile(IAdsUser userIAds)

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want a partial profile.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The hierarchical user object that represents the user profile.

Discussion

This method creates a new user profile with values set for the Cims and User properties. If the zone is an SFU zone, then this method also sets a value for the NISDomain property. You can then add other properties to the profile.

The profile is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

If you pass a null parameter, AddUserPartialProfile throws the exception ArgumentNullException.

Example

The following code sample illustrates using the AddUserPartialProfile method in a script:

```
...
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Note: There is no cims.connect function.
// By default, this application will use the connection to the domain controller

// and existing credentials from the computer already logged in.
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
   return;
}
IUser objUser = cims.GetUserByPath(strUser);
if (objUser == null)
{
```

```
    Console.WriteLine("User " + strUser + " does not exist.");
    return;
}
IHierarchicalUser objUserUnixProfile = (IHierarchicalUser)
objZone.GetUserUnixProfile(objUser);
if (objUserUnixProfile == null)
{
    // New user for the zone
    objUserUnixProfile = objZone.AddUserPartialProfile(strUser);
}
IRole objRole = objZone.GetRole(strRole);
if (objRole == null)
{
    Console.WriteLine("Role " + strRole + " does not exist.");
    return;
}
IRoleAssignment asg = objUserUnixProfile.GetUserRoleAssignment(objRole);
if (asg != null)
{
    Console.WriteLine("Assignment already exist.");
    return;
}
else
{
    // assigning role to user
    asg = objUserUnixProfile.AddUserRoleAssignment();
    asg.Role = objZone.GetRole(strRole);
    asg.Commit();
}
…
```

CreateCommand

Creates a command right for the zone.

Syntax

ICommand CreateCommand ()

Return value

A command right for the zone.

Discussion

A command right controls who has permission to run a specific command in a zone.

The profile is not stored in Active Directory until you call the Unexpected Link Text method.

Example

The following code sample illustrates using the CreateCommand method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
   ICommand objCmd = objZone.GetCommand(strCmd);
   if (objCmd != null)
   {
      Console.WriteLine("Command " + strCmd + " already exists.");
   }
   else
   {
      objCmd = objZone.CreateCommand();
      objCmd.Name = strCmd;
      objCmd.CommandPattern = strPattern;
      objCmd.Description = "optional description";
      objCmd.Commit();
      Console.WriteLine("Command " + strCmd + " was created successfully.");
   }
}
...
```

CreateNetworkAccess

Creates a network application access right.

Syntax

INetworkAccess CreateNetworkAccess ()

Return value

A network application access right for the zone.

Discussion

A network access right enables a user to run an application on a remote computer as another user. For example, a network access right can give a user the ability to run as an SQL Administrator on a remote server.

The right is not stored in Active Directory until you call the Unexpected Link Text method.

Example

The following code sample illustrates using the CreateNetworkAccess method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
   INetworkAccess objNetworkAccess = objZone.GetNetworkAccess(strName);
   if (objNetworkAccess != null)
   {
      Console.WriteLine("NetworkAccess " + strName + " already exist.");
   }
   else
   {
      objNetworkAccess = objZone.CreateNetworkAccess();
      objNetworkAccess.Name = strName;
      objNetworkAccess.RunAsType = WindowsRunAsType.User;
      objNetworkAccess.Priority = 0;
      objNetworkAccess.Description = "optional description";
      string userPath = DirectoryServices.GetLdapPathFromDN(cims.Server, strUser);
      DirectoryEntry userEntry = DirectoryServices.GetDirectoryEntry(userPath,
      cims.UserName, cims.Password);
      SecurityIdentifier m_userSid = new
      SecurityIdentifier(DirectoryServices.GetStringSid(userEntry));
      objNetworkAccess.RunAsList = new List\<SecurityIdentifier\>{ m_userSid };
      objNetworkAccess.Commit();
      Console.WriteLine("NetworkAccess " + strName + " is created successfully.");
   }
}
...
```

CreatePamAccess

Creates a PAM application access right.

Syntax

IPam CreatePamAccess ()

Return value

A PAM application access right for the zone.

Discussion

A PAM (Pluggable Authentication Module) application right gives a user the ability to access the authorized PAM-enabled application.

The right is not stored in Active Directory until you call the 'Commit` method.

Example

The following code sample illustrates using the CreatePamAccess method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
    Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
    IPam objPam = objZone.GetPamAccess(strName);
    if (objPam != null)
    {
    Console.WriteLine("PAM " + strName + " already exists.");
    }
    else
    {
      objPam = objZone.CreatePamAccess();
      objPam.Name = strName;
      objPam.Application = strApp;
      objPam.Description = "optional description";
      objPam.Commit();
    }
}
...
```

CreateRole

Creates a role in the zone.

Syntax

```
IRole CreateRole (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the role.

Return value

A role with the specified name.

Discussion

The role is not stored in Active Directory until you call the Unexpected Link Text method.

Example

The following code sample illustrates using the CreateRole method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

// create the role
if (objZone == null)
{
    Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
    IRole role = objZone.CreateRole(strRole);
    role.Description = "optional description";
    role.Commit();
}
...
```

CreateSshRight

Creates an SSH application access right.

Syntax

```
ISsh CreateSshRight ()
```

Return value

An SSH application access right for the zone.

Discussion

An SSH (Secure Shell) application right gives a user the ability to access the authorized SSH-enabled application.

The right is not stored in Active Directory until you call the Unexpected Link Text method.

CreateWindowsApplication

Creates a Windows application access right.

Syntax

```
IWindowsApplication CreateWindowsApplication ()
```

Return value

A Windows application access right for the zone.

Discussion

A Windows application right gives a user the ability to access the authorized Windows application.

The right is not stored in Active Directory until you call the Unexpected Link Text method.

Example

The following code sample illustrates using the CreateWindowsApplication method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
    Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
    IWindowsApplication objWindowsApplication =
    objZone.GetWindowsApplication(strName);
    if (objWindowsApplication != null)
    {
        Console.WriteLine("WindowsApplication " + strName + " already exists.");
    }
    else
    {
        objWindowsApplication = objZone.CreateWindowsApplication();
        objWindowsApplication.Name = strName;
        objWindowsApplication.RunAsType = WindowsRunAsType.Self;
        objWindowsApplication.Priority = 0;
        objWindowsApplication.Description = "optional description";
        objWindowsApplication.Command = strApplication;
        objWindowsApplication.Commit();
        Console.WriteLine("Windows Application " + strName + " has been created
            successfully.");
    }
}
...
```

CreateWindowsDesktop

Creates a Windows Desktop access right.

Syntax

```
IWindowsDesktop CreateWindowsDesktop ()
```

Return value

A Windows desktop access right for the zone.

Discussion

A Windows desktop right provides a complete desktop that behaves as if the user had logged in as specific privileged user. For example, if you have an SQL Administrator login, you can give an ordinary user an SQL Administrator desktop so they can operate in that role when necessary.

The right is not stored in Active Directory until you call the Unexpected Link Text method.

Example

The following code sample illustrates using the CreateWindowsDesktop method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
   IWindowsDesktop objWindowsDesktop = objZone.GetWindowsDesktop(strName);
   if (objWindowsDesktop != null)
   {
      Console.WriteLine("WindowsDesktop " + strName + " already exists.");
   }
   else
   {
      objWindowsDesktop = objZone.CreateWindowsDesktop();
      objWindowsDesktop.Name = strName;
      objWindowsDesktop.RunAsType = WindowsRunAsType.Self;
      objWindowsDesktop.Priority = 0;
      objWindowsDesktop.Description = "optional description";
      objWindowsDesktop.Commit();
      Console.WriteLine("Windows Desktop " + strName + " has been created
          successfully.");
}
}
...
```

GeneratePredefinedRights

Generates predefined SSH and PAM rights in this zone.

Syntax

```
Void GeneratePredefinedRights ()
```

Discussion

This method calls the Unexpected Link Text and Unexpected Link Text methods for a predefined list of SSH and PAM applications. You can call the Unexpected Link Text and Unexpected Link Text methods to get lists of the SSH and PAM rights that have been created in the zone. The rights are stored in Active Directory; you do not have to call the Unexpected Link Text method.

GeneratePredefinedRoles

Generates predefined roles.

Syntax

Void GeneratePredefinedRoles ()

Discussion

This method calls the Unexpected Link Text method for a predefined list of user roles, such as the Windows Login and UNIX Login roles. You can call the Unexpected Link Text method to get a list of the roles that have been created in the zone. The roles are stored in Active Directory; you do not have to call the Unexpected Link Text ` method.

GetAccessGroup

Gets a user group assigned to this zone given a specific role.

Syntax

```
IHzRoleAssignment GetAccessGroup(IRole role, DirectoryEntry group)
IHzRoleAssignment GetAccessGroup(IRole role, SearchResult groupSr)
IHzRoleAssignment GetAccessGroup(IRole role, string groupDn)
IHzRoleAssignment GetAccessGroup(IRole role, IADsGroup groupIAds)
```

Parameters

Specify the following parameter when using this method:

role	The role of the group.

Specify one of the following parameters when using this method.

group	The directory entry for the group.
groupSr	The directory entry for the group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The computer role assignment that includes the specified group (IHzRoleAssignment.TrusteeType==Group).

Discussion

Any number of user groups can be assigned to a computer role and each of those groups can have more than one role. Use this method to get the computer role assignment for a specific group and role.

The GetAccessGroup(IRole role, DirectoryEntry group) and GetAccessGroup(IRole role, SearchResult groupSr) methods are available only for .NET-based programs; call Unexpected Link Text VBScript.

Exceptions

GetAccessGroup may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid user; or if it failed to create a role assignment because it cannot find the user.

Example

See Unexpected Link Text for an example of using the GetAccessGroup method in a script:

GetAccessGroups

Returns the computer roles assigned to this zone.

Syntax

```
IRoleAssignments GetAccessGroups()
```

Return value

The collection of computer roles. Enumerate this object to get all of the `IAzRoleAssignment` objects in this zone.

GetChildZones

Returns the child zones of this zone.

Syntax

```
IEnuerable GetChildZones()
```

Return value

The collection of child zones.

Exceptions

`GetChildZones` throws an `ApplicationException` if it can't find the zone. For example, `GetChildZones` throws an `ApplicationException` if the zone has been removed or the server is not available.

GetCommand

Returns the command right with a specified name or GUID.

Syntax

ICommand GetCommand (string name)

ICommand GetCommand (Guid id)

Parameter

Specify one of the following parameters when using this method:

name	The name of the command.
id	The GUID of the command.

Return value

A command right with the specified name or GUID, or null if no match is found.

Exceptions

GetCommand may throw one of the following exceptions:

- ApplicationException if it can't find authorization data for the zone or if it failed to get the command right (see the message returned by the exception for the reason).
- ArgumentException if the name or id parameter is null or empty.

Example

The following code sample illustrates using the GetCommand method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
   ICommand objCmd = objZone.GetCommand(strCmd);
   if (objCmd != null)
   {
      Console.WriteLine("Command " + strCmd + " already exists.");
   }
   else
   {
      objCmd = objZone.CreateCommand();
      objCmd.Name = strCmd;
      objCmd.CommandPattern = strPattern;
      objCmd.Description = "optional description";
      objCmd.Commit();
   }
}
...
```

GetCommands

Returns all the command rights in the zone.

Syntax

```
ICommands GetCommands()
```

Return value

The collection of commands in the zone.

GetComputerRole

Returns the computer role with a specified name.

Syntax

IComputerRole GetComputerRole (string name)

Parameter

Specify the following parameter when using this method:

name	The name of the computer role.

Return value

The computer role with the specified name, or null if no match is found.

Exceptions

GetComputerRole throws an ApplicationException if it can't find authorization data for the zone or if it failed to get the computer role (see the message returned by the exception for the reason).

Example

The following code sample illustrates using the GetComputerRole method in a script:

```
...
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
IComputerRole compRole = objZone.GetComputerRole(strName);
if (compRole != null)
{
Console.WriteLine("Computer role " + strName + " already exist.");
}
else
{
compRole = objZone.AddComputerRole(strName);
compRole.Group = strGroup;
compRole.Validate();
compRole.Commit();
Console.WriteLine("Computer role " + strName + " is created successfully.");
}
}
...
```

GetComputerRoles

Returns all the computer roles in the zone.

Syntax

```
IComputerRoles GetComputerRoles()
```

Return value

The collection of computer roles in the zone.

GetEffectiveCommands

Returns all the command rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
ICommands GetEffectiveCommands()
```

Return value

The collection of effective command rights in the zone.

Exceptions

`GetEffectiveCommands` throws an `ApplicationException` if it failed to get the effective command rights (see the message returned by the exception for the reason).

GetEffectiveNetworkAccesses

Returns all the network access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
INetworkAccesses GetEffectiveNetworkAccesses()
```

Return value

The collection of effective network access rights in the zone.

Exceptions

`GetEffectiveNetworkAccesses` throw an `ApplicationException` if it failed to get the effective network access rights (see the message returned by the exception for the reason).

GetEffectivePamAccesses

Returns all the PAM application access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
IPams GetEffectivePamAccesses()
```

Return value

The collection of effective PAM application access rights in the zone.

Exceptions

`GetEffectivePamAccesses` throws an `ApplicationException` if it failed to get the effective command rights (see the message returned by the exception for the reason).

GetEffectiveRoles

Returns all the roles that can be assigned to users in the zone, including roles inherited from zones higher in the hierarchy.

Syntax

```
IRoles GetEffectiveRoles()
```

Return value

The collection of effective roles in the zone.

Exceptions

`GetEffectiveRoles` throws an `ApplicationException` if it failed to get the effective command rights (see the message returned by the exception for the reason).

GetEffectiveSshs

Returns all the SSH application access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
ISshs GetEffectiveSshs()
```

Return value

The collection of effective SSH application access rights in the zone.

Exceptions

`GetEffectiveSshs` throws an `ApplicationException` if it fails to get the effective command rights (see the message returned by the exception for the reason).

GetEffectiveUserUnixProfiles

Returns all the users in the zone, including users inherited from zones higher in the hierarchy.

Syntax

```
IUserUnixProfiles GetEffectiveUserUnixProfiles()
```

Return value

The collection of effective users in the zone.

GetEffectiveWindowsApplications

Returns all the Windows application access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
IWindowsApplications GetEffectiveWindowsApplications()
```

Return value

The collection of effective Windows application access rights in the zone.

Exceptions

`GetEffectiveWindowsApplications` throws an `ApplicationException` if it fails to get the effective access rights (see the message returned by the exception for the reason).

GetEffectiveWindowsDesktops

Returns all the Windows desktop access rights that can be assigned to users in the zone, including rights inherited from zones higher in the hierarchy.

Syntax

```
IwindowsDesktops GetEffectiveWindowsDesktops()
```

Return value

The collection of effective Windows desktop access rights in the zone.

Exceptions

`GetEffectiveWindowsDesktops` throws an `ApplicationException` if it failed to get the effective access rights (see the message returned by the exception for the reason).

GetEffectiveWindowsUsers

Returns all the Windows users in the zone, including users inherited from zones higher in the hierarchy.

Syntax

```
IWindowsUsers GetEffectiveWindowsUsers()
```

Return value

The collection of effective Windows users in the zone.

GetNetworkAccess

Returns the specified network access right.

Syntax

```
INetworkAccess GetNetworkAccess (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the access right.

Return value

The network access right with the specified name.

Exceptions

GetNetworkAccess may throw one of the following exceptions:

- ArgumentException if the parameter value is null or empty.
- ApplicationException if cannot find authorization for the zone, or if it failed to get the network access right (see the message returned by the exception for the reason).

Example

For an example of the use of the GetNetworkAccess method, see Unexpected Link Text.

GetNetworkAccesses

Returns all the network access rights that can be assigned to users in the zone.

Syntax

```
INetworkAccesses GetNetworkAccesses()
```

Return value

The collection of NSS variable names.

Discussion

This method returns only the network access rights assigned in the current zone. Call `GetEffectiveNetworkAccesses` to return all the network access rights including those inherited from zones higher in the hierarchy.

GetNSSVariable

Returns the specified NSS environment variable; VBScript only.

Syntax

```
string GetNssVariable (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the variable.

Return value

The value of the variable, or null if name is not a defined variable.

GetNSSVariables

Returns the names of all NSS variables; VBScript only.

Syntax

```
IEnumerable GetNSSVariables()
```

Return value

The collection of NSS variable names.

GetPamAccess

Returns the specified PAM application access right.

Syntax

```
IPam GetPamAccess (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the PAM access right.

Return value

The PAM application access right with the specified name, or null if name is not in use.

Exceptions

GetPamAccess may throw one of the following exceptions:

- ApplicationException if it can't find authorization data for the zone or if it failed to get the PAM application access right (see the message returned by the exception for the reason).
- ArgumentException if the name parameter is null or empty.

Example

For sample code using the GetPamAccess method in a script, see Unexpected Link Text.

GetPamAccesses

Returns all the PAM application access rights in the zone.

Syntax

```
IPams GetPamAccesses()
```

Return value

The collection of PAM application access rights in the zone.

GetPrimaryUser

Returns the primary profile for the specified user.

Syntax

```
IHierarchicalUser GetPrimaryUser(DirectoryEntry userDE)
IHierarchicalUser GetPrimaryUser(SearchResult userSR)
IHierarchicalUser GetPrimaryUser(string userDn)
IHierarchicalUser GetPrimaryUser(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the primary profile.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The hierarchical user object that represents the user profile.

Discussion

The primary profile is the profile at the highest level in the zone hierarchy where the user's profile is defined. All or part of the primary profile can be overridden by secondary profiles farther down in the hierarchy.

The `GetPrimaryUser(DirectoryEntry userDE)` and `GetPrimaryUser(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

`GetPrimaryUser` throws an `ArgumentNullException` if the specified parameter value is `null` or empty, or if the user does not exist.

GetRole

Returns the role with a specified name or GUID.

Syntax

IRole GetRole (string name)

IRole GetRole (Guid id)

Parameter

Specify the following parameter when using this method:

name	The name of the role.
id	The GUID of the role.

Return value

The role with the specified name, or null if no match is found.

Exceptions

GetRole may throw one of the following exceptions:

- ApplicationException if it can't find authorization data for the zone or if it failed to get the role (see the message returned by the exception for the reason).
- ArgumentException if the parameter is null or empty.

Example

The following code sample illustrates using the GetRole method in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
    Console.WriteLine("Zone " + strZone + " does not exist.");
    return;
}
IRole role = objZone.GetRole(strRole);
if (role == null)
{
    Console.WriteLine(strRole + " does not exist in zone.");
}
else if (objZone.GetAccessGroup(role, strGroup) != null)
{
    Console.WriteLine("Role assignment already exist.");
}
else
{
    // assign a role to the group
    IRoleAssignment zag = objZone.AddAccessGroup(strGroup);
    zag.Role = role;
    zag.Commit();
}
...
```

GetRoleAssignment

Returns a role assignment given a role and trustee.

Syntax

```
IRoleAssignment GetRoleAssignment (IRole role, string dn)
```

Parameter

Specify the following parameters when using this method.

role	The role for which you want the assignment.
dn	The distinguished name of the user or group to whom the role is assigned.

Return value

The role assignment, or `null` if no match is found.

Exceptions

`GetRoleAssignment` throws an `ArgumentNullException` if either parameter is `null` or empty.

GetRoleAssignmentById

Returns a role assignment given an ID.

Syntax

```
IRoleAssignment GetRoleAssignmentById (Guid id)
```

Parameter

Specify the following parameter when using this method:

id	The GUID of the role assignment.

Return value

The role assignment, or `null` if no match is found.

Exceptions

`GetRoleAssignment` throws an `ArgumentException` if the parameter is empty.

GetRoleAssigments

Returns all the role assignments in the zone.

Syntax

```
IRoleAssignments GetRoleAssignments()
```

Return value

The collection of role assignments in the zone.

GetRoleAssignmentToAllADUsers

Returns the role assignment given to all Active Directory users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToAllADUsers(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignments.

Return value

The role assignment for the specified role.

Exceptions

GetRoleAssignmentToAllADUsers throws an ArgumentNullException if the parameter is null.

GetRoleAssignmentToAllUnixUsers

Returns the role assignment given to all UNIX users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToAllUnixUsers(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignments.

Return value

The role assignment for the specified role.

Discussion

This method returns the role assignment for local UNIX users with the specified role.

Exceptions

GetRoleAssignmentToAllUnixUsers throws an ArgumentNullExceptionif the parameter is null.

GetRoles

Returns all the roles in the zone.

Syntax

```
IRoles GetRoles()
```

Return value

The collection of computer roles in the zone.

GetSecondaryUsers

Returns the secondary profiles for the specified user.

Syntax

```
IUserUnixProfiles GetSecondaryUsers(DirectoryEntry userDE)
IUserUnixProfiles GetSecondaryUsers(SearchResult userSR)
IUserUnixProfiles GetSecondaryUsers(string userDn)
IUserUnixProfiles GetSecondaryUsers(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the secondary profiles.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The collection of secondary user UNIX profiles.

Discussion

The primary profile is the profile at the highest level in the zone hierarchy where the user's profile is defined. All or part of the primary profile can be overridden by secondary profiles farther down in the hierarchy.

The `GetSecondaryUser(DirectoryEntry userDE)` and `GetSecondaryUser(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

`GetSecondaryUsers` throws an `ArgumentNullException` if the parameter is `null` or the user does not exist.

GetSshRight

Returns the specified SSH application access right.

Syntax

```
ISsh GetSshRight (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the SSH access right.

Return value

The SSH application access right with the specified name, or `null` if name is not in use.

Exceptions

`GetSshRight` may throw one of the following exceptions:

- `ApplicationException` if it can't find authorization data for the zone or if it failed to get the right (see the message returned by the exception for the reason).
- `ArgumentException` if the parameter is `null` or empty.

GetSshRights

Returns all the SSH application access rights in the zone.

Syntax

```
ISshs GetSshRights()
```

Return value

The collection of SSH application access rights in the zone.

GetSubTreeRoleAssignments

Returns all the role assignments under this zone, including role assignments for computer roles and computers.

Syntax

```
IRoleAssignments GetSubTreeRoleAssignments()
```

Return value

The collection of role assignments in the zone.

Exceptions

`GetSubTreeRoleAssignments` throws an `ApplicationException` if the method fails to find the authorization store.

GetUserProfiles

Returns all the profiles for the specified user.

Syntax

```
IUserUnixProfiles GetUserProfiles(DirectoryEntry userDE)
IUserUnixProfiles GetUserProfiles(SearchResult userSR)
IUserUnixProfiles GetUserProfiles(string userDn)
IUserUnixProfiles GetUserProfiles(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the profiles.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The collection of user UNIX profiles.

Exceptions

GetUserProfiles throws an ArgumentNullException if the parameter is null or the user does not exist.

GetUserRoleAssignments

Returns all the user role assignments in the zone, or for a specific user in the zone.

Syntax

```
IRoleAssignments GetUserRoleAssignments()
IRoleAssignments GetUserRoleAssignments(DirectoryEntry userDE)
IRoleAssignments GetUserRoleAssignments(SearchResult userSR)
IRoleAssignments GetUserRoleAssignments(string userDn)
IRoleAssignments GetUserRoleAssignments(IAdsUser userIAds)
IRoleAssignments GetUserRoleAssignments(IUser user)
```

Parameters

Specify no parameters to return all the role assignments in the zone.

Specify one of the following parameters to return all the role assignments for a specific user:

userDE	The directory entry for the user for which you want the role assignments.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.
user	The user specified as a CIMS user object.

Return value

The collection of role assignments as `IRoleAssignment` objects.

Exceptions

`GetUserRoleAssignments` throws an `ArgumentNullException` if the required parameter is `null` or empty.

GetWindowsApplication

Returns the specified Windows application right.

Syntax

```
IWindowsApplication GetWindowsApplication (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the Windows application.

Return value

The Windows application right with the specified name, or `null` if name is not in use.

Exceptions

`GetWindowsApplication` may throw one of the following exceptions:

- `ApplicationException` if it can't find authorization data for the zone or if it failed to get the right (see the message returned by the exception for the reason).
- `ArgumentException` if the parameter is null or empty.

GetWindowsApplications

Returns all the Windows application rights in the zone.

Syntax

```
IWindowsApplications GetWindowsApplications()
```

Return value

The collection of Windows application rights in the zone.

GetWindowsComputers

Returns all the Windows computers in the zone.

Syntax

```
IComputers GetWindowsComputers()
```

Return value

The collection of Windows computers in the zone.

GetWindowsDesktop

Returns the specified Windows desktop right.

Syntax

```
IWindowsDesktop GetWindowsDesktop (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the Windows desktop.

Return value

The Windows desktop right with the specified name, or `null` if name is not in use.

Exceptions

`GetWindowsDesktop` may throw one of the following exceptions:

- `ApplicationException` if it can't find authorization data for the zone or if it failed to get the right (see the message returned by the exception for the reason).
- `ArgumentException` if the parameter is null or empty.

GetWindowsDesktops

Returns all the Windows desktop rights in the zone.

Syntax

```
IWindowsDesktops GetWindowsDesktops()
```

Return value

The collection of Windows desktop rights in the zone.

PrecreateComputerZone

Adds a computer zone to a computer object in this zone.

Syntax

```
IHierarchicalZoneComputer PrecreateComputerZone(string dnsname, DirectoryEntry trustee)
IHierarchicalZoneComputer PrecreateComputerZone(string dnsname, string trusteeDn)
IHierarchicalZoneComputer PrecreateComputerZone (string dnsName, DirectoryEntry trustee, bool skipPermissionSetting);
```

Parameters

Specify the following parameters when using this method.

dnsname	The DNS host name of the Active Directory computer object to which you wish to add a computer zone.
trustee	The user or group to which the computer-level overrides will be assigned.
trusteeDn	The user or group to which the computer-level overrides will be assigned, specified as a distinguished name.
skipPermissionSetting	Specifies if permission delegation is skipped when precreating computer zones.

Return value

The hierarchical computer object that contains the computer zone.

Discussion

Computer-level overrides for user, group, or computer role assignments are contained in a *computer zone*, which is a special type of zone that contains properties that are specific to only one computer. Computer zones are an internal data structure that are not exposed as zone in Access Manager.

This method adds a computer zone to a computer object in a hierarchical zone. You can then assign roles to that trustee.

Use `PrecreateComputerZone(string dnsname, DirectoryEntry trustee)` for .NET programs and `PrecreateComputerZone(string dnsname, string trusteeDn)` for VBScript.

Exceptions

`PrecreateComputerZone` throws an `ApplicationException` if the method fails to delegate computer zone permissions (see the message returned by the exception for the reason).

SetNSSVariable

Sets the value of the specified NSS environment variable; VBScript only.

Syntax

string SetNssVariable (string name, string value)

Parameter

Specify the following parameters when using this method.

name	The name of the variable.
value	The value of the variable.

GroupDefaultName

Gets or sets the default name for new group profiles.

Syntax

```
string GroupDefaultName {get; set;}
```

Property value

The default group profile name.

IsChild

Indicates whether this is a child zone.

Syntax

```
bool IsChild {get;}
```

Property value

Returns `true` if this zone is designated a child zone.

Discussion

Delinea allows a child zone to have no parent, so that you can preload all the child zone profiles and role assignments before assigning the zone to a parent zone. This property identifies a zone as a parent zone if it has no link to a parent zone and has a child zone pointing to it. If the zone is not a parent zone according to those criteria, then this property returns `true`, identifying it as a child zone.

IsGroupDefaultNameDefined

Determines whether the group default name has been defined.

Syntax

```
bool IsGroupDefaultNameDefined {get; set;}
```

Property value

Returns `true` if a group default name has been defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Exceptions

`IsGroupDefaultNameDefined` throws an `InvalidOperationException` if the group default name has not been defined and you attempt to set this property `true`.

IsNextGidDefined

Determines whether the NextGID property has been defined for this zone.

Syntax

```
bool IsNextGidDefined {get; set;}
```

Property value

Returns true if the NextGID property has been defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Exceptions

IsNextGidDefined throws an InvalidOperationException if the NextGID property has not been defined and you attempt to set this property true.

IsNextUidDefined

Determines whether the NextUID property has been defined for this zone.

Syntax

```
bool IsNextUidDefined {get; set;}
```

Property value

Returns true if the Unexpected Link Text property has been defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Exceptions

IsNextUidDefined throws an InvalidOperationException if the NextUID property has not been defined and you attempt to set this property true.

IsUseAutoPrivateGroupDefined

Determines whether the UseAutoPrivateGroup property has been defined for this zone.

Syntax

```
bool IsUseAutoPrivateGroupDefined {get; set;}
```

Property value

Returns true if the UseAutoPrivateGroup property has been defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Discussion

When auto private groups are enabled, the user's UNIX profile name is automatically used as the group name and the user's UID is used as the GID.

Exceptions

IsUseAutoPrivateGroupDefined throws an InvalidOperationException if the UseAutoPrivateGroup property has not been defined and you attempt to set this property true.

IsUserDefaultGecosDefined

Determines whether the UserDefaultGecos property has been defined for this zone.

Syntax

```
bool IsUserDefaultGecosDefined {get; set;}
```

Property value

Returns true if the UserDefaultGecos property has been defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Exceptions

IsUserDefaultGecosDefined throws an InvalidOperationException if the UserDefaultGecos property has not been defined and you attempt to set this property true.

IsUserDefaultHomeDirectoryDefined

Determines whether the UserDefaultHomeDirectory property is defined for this zone.

Syntax

```
bool IsUserDefaultHomeDirectoryDefined {get; set;}
```

Property value

Returns true if the UserDefaultHomeDirectory property is defined. Set this value false to clear this property. Call the Unexpected Link Text property to set a value for this property.

Exceptions

IsUserDefaultHomeDirectoryDefined throws an InvalidOperationException if the UserDefaultHomeDirectory property has not been defined and you attempt to set this property true.

IsUserDefaultNameDefined

Determines whether the UserDefaultName property is defined for this zone.

Syntax

```
bool IsUserDefaultNameDefined {get; set;}
```

Property value

Returns true if the user default name property is defined. Set this value false to clear this property. Call the UserDefaultName property to set a value for this property.

Exceptions

IsUserDefaultNameDefined throws an InvalidOperationException if the UserDefaultName property has not been defined and you attempt to set this property true.

IsUserDefaultPrimaryGroupDefined

Determines whether the UserDefaultPrimaryGroup property is defined for this zone.

Syntax

```
bool IsUserDefaultPrimaryGroupDefined {get; set;}
```

Property value

Returns true if the user default primary group property is defined. Set this value false to clear this property. Call the UserDefaultPrimaryGroup property to set a value for this property.

Exceptions

IsUserDefaultPrimaryGroupDefined throws an InvalidOperationException if the UserDefaultPrimaryGroup property has not been defined and you attempt to set this property true.

IsUserDefaultRoleDefined

Determines whether the UserDefaultRole property is defined for this zone.

Syntax

```
bool IsUserDefaultRoleDefined {get; set;}
```

Property value

Returns true if the user default role property is defined. Set this value false to clear this property. Call the UserDefaultRole property to set a value for this property.

Exceptions

IsUserDefaultRoleDefined throws an InvalidOperationException if the UserDefaultRole property has not been defined and you attempt to set this property true.

IsUserDefaultShellDefined

Determines whether the UserDefaultShell property is defined for this zone.

Syntax

```
bool IsUserDefaultShellDefined {get; set;}
```

Property value

Returns true if the user default shell property is defined. Set this value false to clear this property. Call the DefaultShell property to set a value for this property.

Exceptions

IsUserDefaultShellDefined throws an InvalidOperationException if the UserDefaultShell property has not been defined and you attempt to set this property true.

NssVariables

Gets all the NSS environment variables.

Syntax

IDictionary\<string, string\> NssVariables {get;}

Property value

A dictionary of key-value pairs that define all the profile variables.

Discussion

The NSSVariables property is available only for .NET-based programs; call Unexpected Link Text for VBScript.

Example

The following code sample illustrates using the NssVariables property in a script:

```
...
IHierarchicalZone objParent =
cims.GetZoneByPath("cn=" + strParentZone + "," + strContainerDN) as
IHierarchicalZone;
if (objParent == null)
{
    Console.WriteLine("Parnet zone " + strParentZone + " does not exist.");
}
else
{
    IHierarchicalZone objZone = cims.CreateZone(objContainer, strZone) as IHierarchicalZone;
    // set the starting UID and GID for the zone
    objZone.NextUID = 10000;
    objZone.NextGID = 10000;
    objZone.UseNextUid = true;
    objZone.UseNextGid = true;
    objZone.AvailableShells = new string[] { "/bin/bash", "/bin/shell" };
    objZone.DefaultShell = "%{shell}";
    objZone.DefaultHomeDirectory = "%{home}/%{user}";
    objZone.UserDefaultGecos = "%{u:description}";
    objZone.Parent = objParent;
    objZone.NssVariables.Add("shell", "/bin/bash" );
    objZone.Commit();
}
...
```

Parent

Gets or sets the parent of the current zone.

Syntax

```
IHierarchicalZone Parent {get; set;}
```

Property value

The parent zone.

Discussion

Delinea allows a child zone to have no parent, so that you can preload all the child zone profiles and role assignments before assigning the zone to a parent zone. See also the discussion under the Unexpected Link Text property.

SFU zones cannot be child zones. See Data storage for Delinea zones for information about different zone types.

Exceptions

Parent throws an ApplicationException if you attempt to assign a parent to an SFU zone.

Example

The following code sample illustrates using the Parent property in a script:

```
...
IHierarchicalZone objParent =
cims.GetZoneByPath("cn=" + strParentZone + "," + strContainerDN) as
IHierarchicalZone;
if (objParent == null)
{
    Console.WriteLine("Parent zone " + strParentZone + " does not exist.");
}
else
{
    IHierarchicalZone objZone = cims.CreateZone(objContainer, strZone) as IHierarchicalZone;
    // set the starting UID and GID for the zone
    objZone.NextUID = 10000;
    objZone.NextGID = 10000;
    objZone.UseNextUid = true;
    objZone.UseNextGid = true;
    objZone.AvailableShells = new string[] { "/bin/bash", "/bin/shell" };
    objZone.DefaultShell = "%{shell}";
    objZone.DefaultHomeDirectory = "%{home}/%{user}";
    objZone.UserDefaultGecos = "%{u:description}";
    objZone.Parent = objParent;
    objZone.NssVariables.Add("shell", "/bin/bash" );
    objZone.Commit();
}
...
```

UseAppleGid

Determines whether to use the Apple algorithm to automatically generate the GID when adding a group to the zone. The Apple algorithm uses the globally unique identifier (GUID) for the object to generate a unique numeric identifier for each group.

Syntax

```
bool UseAppleGid {get; set;}
```

Property value

If this value is set to true, Delinea uses the Apple algorithm to generate the numeric group identifier (GID) when adding a group.

UseAppleUid

Determines whether to use the Apple algorithm to automatically generate the UID when adding a user to the zone. The Apple algorithm uses the globally unique identifier (GUID) for the object to generate a unique numeric identifier for each user.

Syntax

```
bool UseAppleUid {get; set;}
```

Property value

If this value is set to true, Delinea uses the Apple algorithm to generate the numeric user identifier (UID) when adding a user.

UseAutoGid

Determines whether to use the Delinea algorithm to automatically generate the GID when adding a group to the zone.an auto-generated GID when adding a group to the zone. The Delinea algorithm uses the domain-wide security identifier (SID) and the relative identifier (RID) to generate a unique numeric identifier for each group.

Syntax

```
bool UseAutoGid {get; set;}
```

Property value

If this value is set to true, Delinea automatically generates a GID from the domain-wide security identifier (SID) and the relative identifier (RID) when adding a group.

UseAutoPrivateGroup

Determines whether to use a private group when adding a user to the zone.

Syntax

```
bool UseAutoPrivateGroup {get; set;}
```

Property value

When this value is `true`, Delinea uses a private group when adding a user to the zone.

Discussion

A private group automatically sets the user's UNIX profile name as the user's primary group name and the user's UID as the user's primary group GID. Automatically-generated groups are not stored or managed in Active Directory.

Call the `IsUseAutoPrivateGroupDefined` property before attempting to get the value for this property.

Exceptions

`UseAutoPrivateGroup` may throw one of the following exceptions:

- `FormatException` if the GID is not a number.
- `InvalidOperationException` if the `UseAutoPrivateGroup` property has not been defined.

UseAutoUid

Determines whether to use the Delinea algorithm to automatically generate the UID when adding a user to the zone.The Delinea algorithm uses the domain-wide security identifier (SID) and the relative identifier (RID) to generate a unique numeric identifier for each user.

Syntax

```
bool UseAutoUid {get; set;}
```

Property value

If this value is set to `true`, Delinea automatically generates a UID from the domain-wide security identifier (SID) and the relative identifier (RID) when adding a user.

UseNextGid

Determines whether to automatically increment the GID when adding a group to the zone.

Syntax

```
bool UseNextGid {get; set;}
```

Property value

When this value is true, Delinea automatically increments the GID when adding a group to the zone.

Discussion

There is no guarantee that the GIDs generated using this method are unique with regard to GIDs in other zones.

Example

For a code sample illustrating the use of this property, see Unexpected Link Text.

UseNextUid

Determines whether to automatically increment the UID when adding a user to the zone.

Syntax

```
bool UseNextUid {get; set;}
```

Property value

When this value is true, Delinea automatically increments the UID when adding a user to the zone.

Discussion

There is no guarantee that the UIDs generated using this method are unique with regard to UIDs in other zones.

Example

For a code sample illustrating the use of this property, see Unexpected Link Text.

UserDefaultGecos

Gets or sets the default GECOS field for new user profiles in the zone.

Syntax

```
string UserDefaultGecos {get; set;}
```

Property value

The contents of the GECOS field.

Example

For a code sample illustrating the use of this property, see Unexpected Link Text.

UserDefaultGid

Gets or sets the default GID (32-bit) for new user profiles in the zone.

Syntax

```
int UserDefaultGid {get; set;}
```

Property value

The GID. If set to -1, Delinea uses auto private group for new user profiles (see Unexpected Link Text).

Discussion

This property returns an error if the default GID has not been defined. Call the Unexpected Link Text property to determine whether the default GID is defined.

This property uses a 32-bit value for use in VBScript scripts. Use the Unexpected Link Text property for 64-bit GIDs.

Exceptions

UserDefaultGid may throw one of the following exceptions:

- InvalidOperationException if you try to get the default GID and it has not been defined.
- FormatException if the default GID value is not valid.

UserDefaultName

Gets or sets the default name for new user profiles in the zone.

Syntax

```
string UserDefaultName {get; set;}
```

Property value

The default name.

UserDefaultPrimaryGroup

Gets or sets the default GID (64-bit) for new user profiles in the zone.

Syntax

```
long UserDefaultPrimaryGroup {get; set;}
```

Property value

The GID. If set to -1, Delinea uses private groups for new user profiles (see Unexpected Link Text).

Discussion

This property returns an error if the default GID has not been defined. Call the Unexpected Link Text property to determine whether the default GID is defined.

This property uses a 64-bit value for use in .NET modules. Use the Unexpected Link Text property for VBScript.

Exceptions

`UserDefaultPrimaryGroup` may throw one of the following exceptions:

- `InvalidOperationException` if you try to get the default GID and it has not been defined.
- `FormatException` if the default GID value is not valid.

UserDefaultRole

Gets or sets the default role for new user profiles in the zone.

Syntax

```
IRole UserDefaultRole {get; set;}
```

Property value

The default role.

Delinea

HzRoleAssignment

Represents a zone-level role assignment.

Syntax

public interface IRoleAssignment

Methods

The HzRoleAssignment class provides the following methods:

Unexpected Link Text	VBScript interface to clear the custom attributes for this class. Inherited from Unexpected Link Text
Unexpected Link Text	Commits changes in the role assignment to Active Directory. Inherited from Unexpected Link Text
Unexpected Link Text	Deletes the role. Inherited from Unexpected Link Text
ICustomAttributeContainer Unexpected Link Text	.NET interface that returns the directory entry for the parent container object for the custom attributes for this class. Inherited from Unexpected Link Text
Unexpected Link Text	Returns the trustee being assigned. Inherited from Unexpected Link Text
Unexpected Link Text	VBScript interface to set the custom attributes for this class. Inherited from Unexpected Link Text
Unexpected Link Text	Validates the role assignment. Inherited from Unexpected Link Text

Properties

The HzRoleAssignment class provides the following properties:

Unexpected Link Text	VBScript only: Gets or sets custom attributes for this class. Inherited from Unexpected Link Text
Unexpected Link Text	Determines the time at which this role becomes inactive. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the GUID of the role assignment. Inherited from Unexpected Link Text
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid role. Inherited from Unexpected Link Text
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid trustee. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the local trustee being assigned. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the role the trustee is assigned to. Inherited from Unexpected Link Text
Unexpected Link Text	Specifies the time from which this role becomes effective. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the distinguished name of the trustee assigned this role. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the trustee type of the role assignment. Inherited from Unexpected Link Text
Unexpected Link Text	Gets the zone in which the role assignment is made.

Zone

Gets the zone in which the role assignment is made.

Syntax

IZone Zone {get;}

Property value

The zone of the role assignment.

InheritedRoleAsg

The InheritedRoleAsg class represents a virtual role assignment constructed by inheritance from parent zones and the current zone or computer. A role assignment object contains information about an Active Directory object (trustee) that has been added to a role.

Syntax

public interface IInheritedRoleAsg

Methods

The InheritedRoleAsg class provides the following method:

Unexpected Link Text	Returns the user associated with the role.

Properties

The InheritedRoleAsg class provides the following properties:

Unexpected Link Text	Determines the time at which this role becomes inactive.
Unexpected Link Text	Indicates whether the role assignment is orphaned due to missing or invalid data.
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing trustee.
Unexpected Link Text	Gets the role the trustee is assigned to.
Unexpected Link Text	Gets the role assignment that is the source for this inherited role assignment.
Unexpected Link Text	Specifies the time from which this role becomes effective.
Unexpected Link Text	Gets the distinguished name of the trustee assigned this role.

GetTrustee

Returns the user associated with the role.

Syntax

```
DirectoryEntry GetTrustee()
```

Return value

The Active Directory object representing the user.

EndTime

Determines the time from which this role becomes inactive.

Syntax

```
DateTime EndTime {get; set;}
```

Property value

The time at which the role ceases to be active. A value of `DateTime.MaxValue` means the role never expires. The time must be later than 1/1/1970 00:00.

IsRoleOrphaned

Indicates whether the role assignment is orphaned due to a missing or invalid role.

Syntax

```
bool IsRoleOrphaned {get;}
```

Property value

Returns `true` if this role assignment is an orphan.

IsTrusteeOrphaned

Indicates whether the role assignment is orphaned due to a missing or invalid user.

Syntax

```
bool IsTrusteeOrphaned {get;}
```

Property value

Returns `true` if this role assignment is an orphan.

Role

Syntax

IRole Role {get;}

Property value

The object representing the role.

Source

Gets the role assignment that is the source for this inherited role assignment.

Syntax

```
IRoleAssignment Source {get;}
```

Property value

The original role assignment that is the source for this inherited role assignment.

StartTime

Gets the time from which this role becomes effective.

Syntax

```
DateTime StartTime {get;}
```

Property value

The time at which the role becomes active. A value of `DateTime.MinValue` means the role becomes effective immediately. The time must be later than 1/1/1970 00:00.

TrusteeDn

Gets the user associated with the role.

Syntax

```
string TrusteeDn {get;}
```

Property value

The distinguished name of the user.

Key

The Key class provides access to individual license key properties.

Syntax

public interface IKey

Discussion

A Key object represents a single license key provided by Delinea. The license key is an encrypted string that encapsulates information such as the license type, number of allowed computers, a serial number, and whether the license is an evaluation or permanent license. For example, a single license key might authorize access for 25 servers for the organization with the serial number defined as 317.

Properties

The Key class provides the following properties:

Unexpected Link Text	Gets the number of licenses provided by a specific key.
Unexpected Link Text	Gets the date a specific license key is set to expire.
Unexpected Link Text	Determines whether the license key is an evaluation license.
Unexpected Link Text	Determines whether the license key being checked is a valid license.
Unexpected Link Text	Gets the serial number of the license key.
Unexpected Link Text	Gets the license type for a specific license key.

Count

Gets the number of licenses provided by a specific license key.

Syntax

```
int Count {get;}
```

Property value

The number of licenses provided in a license key.

Discussion

Each license key specifies the number of workstations or servers for which you have purchased licenses. This property indicates the total number of licenses defined for the key. It does not indicate the number currently in use or available to be used.

Example

The following code sample illustrates using the license key `Count` property in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
wscript.echo "Number of licenses:", objLics.Count
for each objLic in objLics
wScript.Echo "License Type:", objLic.Type
wScript.Echo "Seats:", objLic.Count
wScript.Echo "Used:", objLic.usedCount
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
if objKey.isEval then
wScript.Echo "-- [Eval] ", objKey.ExpiryDate
else
wscript.Echo "--", "Serial \#:", objKey.SerialNumber
wScript.Echo "Seats:", objKey.Count
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

ExpiryDate

Gets the date a specific license key is set to expire.

Syntax

DateTime ExpiryDate {get;}

Property value

The date on which a specified license key expires.

Discussion

If a license key is defined as an evaluation license, it includes a timestamp that determines when the license will expire. You can use this property to retrieve this expiration date.

Example

The following code sample illustrates using ExpiryDate in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
wscript.echo "Number of licenses:", objLics.Count
for each objLic in objLics
wScript.Echo "License Type:", objLic.Type
wScript.Echo "Seats:", objLic.Count
wScript.Echo "Used:", objLic.usedCount
‘Display the expiration for eval licenses
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
if objKey.isEval then
wScript.Echo "-- [Eval] ", objKey.ExpiryDate
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

IsEval

Determines whether the license key is an evaluation license.

Syntax

```
bool IsEval {get;}
```

Property value

Returns `true` if the license is a temporary evaluation license, or `false` if the license key is a permanent license.

Discussion

An evaluation license provides full use of Delinea software for a limited period of time. This property returns `true` if the license is a temporary evaluation license, or `false` if the license key is a permanent license.

Example

The following code sample illustrates using `IsEval` in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
‘Check for evaluation license keys
if objKey.isEval then
wScript.Echo "-- [Eval] ", objKey.ExpiryDate
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

IsValid

Determines whether the license key being checked is a valid license.

Syntax

```
bool IsValid {get;}
```

Property value

Returns `true` if the license is valid, or `false` if the license key is not valid or has expired.

Discussion

This property returns `true` if the license key is a valid license, or `false` if the license key is invalid. The property also returns `false` if the license key checked is an expired evaluation license.

Example

The following code sample illustrates using `IsValid` in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
If not objKey.isValid then
wScript.Echo "Invalid License Key"
wscript.Quit
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

SerialNumber

Gets the serial number of the license key.

Syntax

```
int SerialNumber {get;}
```

Property value

The serial number for a specified license key.

Discussion

The serial number provides a mechanism for tracing which license keys were issued to a recipient.

Example

The following code sample illustrates using SerialNumber in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
if objKey.isValid then
wscript.Echo "--", "Serial \#:", objKey.SerialNumber
wScript.Echo "Seats:", objKey.Count
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

Type

Gets the license type for a specific license key.

Syntax

```
LicenseType Type {get;}
```

Property value

The license type for a license key.

See Unexpected Link Text for possible values.

Discussion

The license type indicates whether a specific license key is intended for workstation computers or application servers.

Example

The following code sample illustrates using `Type` in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
set objKeys = objLic.keys
i = 0
do while i \< objKeys.Count
set objKey = objKeys(0)
if objKey.isValid then
wScript.Echo "License Type", objKey.Type
wscript.Echo "Serial \#:", objKey.SerialNumber
wScript.Echo "Seats:", objKey.Count
end if
i = i + 1
loop
next
wScript.Echo ""
next
...
```

Keys

The Keys class is used to manage a set of license keys.

Syntax

```
public interface IKeys
```

Discussion

This class allows you to retrieve a set of license keys of a particular license type. For example, if you have one or more encrypted license keys (xxxx-xxxx-xxxx) that provide up to 100 licenses of the same license type, such as 100 workstation, server, or application licenses, the Keys object can be used to retrieve the Key objects that provide those 100 workstation, server, or application licenses.

Methods

The Keys class provides the following methods:

Unexpected Link Text	Adds a license key to the set.
Unexpected Link Text	Returns an enumeration of Key objects.
Unexpected Link Text	Removes a license key from the set.

Properties

The Keys class provides the following properties:

Unexpected Link Text	Gets the number of license keys stored in the set.
Unexpected Link Text	Gets the license key object using a specific index identifier.

Add

Adds a license key to the set of keys of a particular type.

Syntax

```
Key Add(string key)
```

Parameter

Specify the following parameter when using this method:

key	The license key string to add to the set.

Return value

The added license key object.

Exceptions

`Add` may throw one of the following exceptions:

- `ApplicationException` if the license key has expired, is invalid, or is a non-FIPS 140 key on a system that requires FIPS 140 keys.
- `ArgumentException` if the parameter is `null` or empty or if the key already exists.

GetEnumerator

Returns an enumeration of Key objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of Key objects.

Remove

Removes a license key from the set of license keys of a particular type.

Syntax

```
void Remove(string key)
```

Parameter

Specify the following parameter when using this method:

key	The license key string to remove from the set.

Return value

The license key string to be removed.

Exceptions

`Remove` may throw one of the following exceptions:

- `ApplicationException` if the license key cannot be found.
- `ArgumentException` if the parameter is null or empty.

Count

Determines the total number of license keys defined in the collection of keys for a particular type of license.

Syntax

```
int Count {get;}
```

Property value

The number of individual license keys included in the Keys collection.

Item

Gets the license key object found at the index point you specify.

Syntax

```
IKey this[int i] {get;}
```

Parameter

Specify the following parameter when using this property.

i	The index number to use for retrieving the license key object from the set.

Property value

The license key object.

Delinea

License

The License class provides access to Delinea license properties.

Syntax

```
public interface ILicense
```

Discussion

This class represents the keys of the same license type in the same license container. For example, the License object can contain the collection of server, workstation, or application licenses for one license container, such as the default parent container domain/Program Data/Centrify/Licenses.

Properties

The License class provides the following properties:

Unexpected Link Text	Determines the total number of licenses of a particular type.
Unexpected Link Text	Determines whether the license is an evaluation license.
Unexpected Link Text	Gets the license keys associated with the license object.
Unexpected Link Text	Gets the license type for the license object.
Unexpected Link Text	Gets the total number of licenses that are currently in use.

Count

Determines the total number of licenses of a particular type.

Syntax

```
int Count {get;}
```

Property value

The number of licenses provided in the License object.

Example

The following code sample illustrates using Count in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
wScript.Echo "License Type:", objLic.Type
wScript.Echo "Seats:", objLic.Count
wScript.Echo "Used:", objLic.usedCount
next
wScript.Echo ""
next
...
```

IsEval

Determines whether the license is an evaluation license.

Syntax

```
bool IsEval {get;}
```

Property value

Returns `true` if the license is a temporary evaluation license, or `false` if it is a permanent license.

Discussion

An evaluation license provides full use of Delinea software for a limited period of time. This property returns `true` if the license is a temporary evaluation license, or `false` if the license is permanent.

Keys

Gets the license keys associated with the license object.

Syntax

```
Keys Keys {get;}
```

Property value

The Keys object that contains the set of individual license keys for this license object.

Type

Gets the license type for the license object.

Syntax

```
LicenseType Type {get;}
```

Property value

The license type.

Possible values:

```
public enum LicenseType
{
    // Not defined
       NotDefined = -1,
    // UNIX Workstation license
       Workstation = 110,
    // UNIX Server license
       Server = 111,
    // Windows Server license
       WindowsServer = 112,
    // Windows Workstation license
       WindowsWorkstation = 113,
    // Application license for Tomcat
       Tomcat = 210,
    // Application license for JBoss
       JBoss = 211,
    // Application license for WebLogic
       WebLogic = 212,
    // Application license for WebSphere
       WebSphere = 213,
    // Application license for Apache
       Apache = 214,
    // Application license for DB2
       DB2 = 215,
    // Install evaluation license
       InstallEval = 310,
    // Specific date evaluation license
       SpecificEval = 311
}
```

Discussion

The license type indicates whether a specific license is intended for workstation computers or application servers.

Example

The following code sample illustrates using `Type` in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
wScript.Echo "License Type:", objLic.Type
wScript.Echo "Seats:", objLic.Count
wScript.Echo "Used:", objLic.usedCount
next
wScript.Echo ""
next
...
```

UsedCount

Gets the total number of licenses that are currently in use.

Syntax

```
int UsedCount {get;}
```

Property value

The total number of licenses that are currently in use.

Discussion

Available licenses become used licenses when computers join the domain.

Exceptions

`UsedCount` throws an `ApplicationException` if license information is unavailable because an LDAP error occurred.

Example

The following code sample illustrates using `UsedCount` in a script:

```
...
set objCollection cims.LoadLicenses
for each objLics in objCollection
for each objLic in objLics
wScript.Echo "License Type:", objLic.Type
wScript.Echo "Seats:", objLic.Count
wScript.Echo "Used:", objLic.usedCount
next
wScript.Echo ""
next
...
```

Licenses

The Licenses class is used to manage a set of licenses in a particular license container object.

Syntax

public interface ILicenses

Discussion

This class represents the collection of License objects that have been added to this parent container object. The Licenses object represents one container in the LicensesCollection object.

Methods

The Licenses class provides the following methods:

Unexpected Link Text	Adds a license key to the set of licenses in the license container.
Unexpected Link Text	Commits any changes to the Licenses object to Active Directory.
Unexpected Link Text	Returns an instance of the directory entry for the Licenses parent container object.
Unexpected Link Text	Reloads the Licenses object data from the data in Active Directory.
Unexpected Link Text	Removes a license key from the set.

Properties

The Licenses class provides the following properties:

Unexpected Link Text	Gets the number of license objects stored in this Licenses container.
Unexpected Link Text	Indicates whether any generated license key installed in the Licenses parent container is an evaluation license.
Unexpected Link Text	Indicates whether any computer license is installed.
Unexpected Link Text	Gets the ID from this Licenses object.
Unexpected Link Text	Indicates whether the Licenses parent object in Active Directory is readable.
Unexpected Link Text	Indicates whether the Licenses parent object in Active Directory is writable.
Unexpected Link Text	Gets the License object for a specific type of license.

AddLicenseKey

Adds a license key to the set of licenses in a particular Licenses parent container.

Syntax

```
IKey AddLicenseKey(string key)
```

Parameter

Specify the following parameter when using this method:

key	The license key string to add to the parent container.

Return value

The added key object.

Exceptions

`AddLicenseKey` may throw one of the following exceptions:

- `ApplicationException` if the license key has expired, is invalid, or is a non-FIPS 140 key on a system that requires FIPS 140 keys.
- `ArgumentException` if the parameter is `null` or empty or if the key already exists.

Commit

Commits any changes or updates to the Licenses object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

The method does not validate the data before saving it in Active Directory.

Exceptions

Commit throws an `ApplicationException` if the Active Directory domain controller is readonly.

GetDirectoryEntry

Returns an instance of the DirectoryEntry object for the Licenses parent container object from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The DirectoryEntry object for the Licenses parent container object.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Refresh

Reloads the Licenses object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the collection of licenses in the cached object to ensure it is synchronized with the latest information in Active Directory.

RemoveLicenseKey

Removes a license key from the set of license objects for a particular Licenses parent container.

Syntax

```
void RemoveLicenseKey(string key)
```

Parameter

Specify the following parameter when using this method:

key	The license key string to remove from the set.

Return value

The license key object to be removed.

Exceptions

RemoveLicenseKey may throw one of the following exceptions:

- ApplicationException if the license key cannot be found.
- ArgumentException if the parameter is null or empty.

Count

Gets the total number of licenses for a particular Licenses parent container.

Syntax

```
int Count {get;}
```

Property value

The number of licenses provided in the Licenses object.

HasEvaluation

Indicates whether any license key installed in the `Licenses` parent container is an evaluation license.

Syntax

```
bool HasEvaluation {get;}
```

Property value

Returns `true` if any generated license key is a temporary evaluation license, or `false` if there are no evaluation license keys installed.

HasMachineLicense

Indicates whether any computer license is installed.

Syntax

```
bool HasMachineLicense {get;}
```

Property value

Returns `true` if any computer license is installed, or `false` if there are no computer licenses installed.

ID

Gets the ID from the Licenses object.

Syntax

```
string ID {get;}
```

Property value

The ID.

IsReadable

Indicates whether the Licenses parent object in Active Directory is readable for the current user credentials.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the Licenses parent object is readable, or false if the object is not readable.

Discussion

This property returns a value of true if the user accessing the Licenses object in Active Directory has sufficient permissions to read its properties.

IsWritable

Indicates whether the Licenses parent object in Active Directory is writable for the current user credentials.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns true if the Licenses parent object is writable, or false if the object is not writable.

Discussion

This property returns a value of true if the user accessing the Licenses object in Active Directory has sufficient permissions to write its properties.

Item

Gets the License object for a specific type of license.

Syntax

```
ILicense this[LicenseType type] {get;}
```

Parameter

Specify the following parameter when using this property.

type	The type of License object to retrieve from the Licenses object.

See Unexpected Link Text for possible values.

Return value

The Unexpected Link Text object of the specified type.

Discussion

This property enables you to retrieve the License object for a collection of licenses of a particular type, such as the collection of server licenses or the collection of licenses for a specific application.

Delinea

LicensesCollection

The LicensesCollection class is used to manage all of the licenses in all of the Licenses parent containers defined for a forest.

Syntax

```
public interface ILicensesCollection
```

Discussion

The LicensesCollection object is retrieved using the Cims.[LoadLicenses](dev/windows-api/object-reference/cims/loadlicenses.md) method.

Methods

The LicensesCollection class provides the following methods:

Unexpected Link Text	Returns the specific parent container object from the collection of all parent license containers in the forest.
Unexpected Link Text	Returns an enumeration of Licenses objects.
Unexpected Link Text	Returns the total number of licenses of the specified license type that have been installed.
Unexpected Link Text	Returns the total number of licenses of the specified license type that are currently being used.

Properties

The LicensesCollection class provides the following properties:

Unexpected Link Text	Gets the number of license container objects stored in the Active Directory forest.
Unexpected Link Text	Indicates whether there are any evaluation licenses in the forest.
Unexpected Link Text	Indicates whether any computer license is installed.
Unexpected Link Text	Gets the parent license container object by index number.

Find

Returns the specific parent container object from the collection of all parent license containers in the forest.

Syntax

```
ILicenses Find(DirectoryEntry licenseContainer)
```

Parameter

Specify the following parameter when using this method:

licenseContainer	The Licenses parent container object to retrieve.

Return value

The specified Licenses container object. The container contains the \$CimsLicenseVersionX object.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

GetEnumerator

Returns an enumeration of Licenses objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of Licenses objects.

GetLicensedCount

Returns the total number of licenses of the specified license type that have been installed in an Active Directory forest.

Syntax

```
int GetLicensedCount(LicenseType type)
```

Parameter

Specify the following parameter when using this method:

type	The license type for which you want to get a complete count.

See Unexpected Link Text for possible values.

Return value

Returns a value that represents the number of licenses installed of the specified type.

GetUsedCount

Returns the total number of licenses of the specified license type that are currently being used in an Active Directory forest.

Syntax

```
int GetUsedCount(LicenseType type)
```

Parameter

Specify the following parameter when using this method:

type	The license type for which you want to determine the number being used.

See Unexpected Link Text for possible values.

Return value

Returns a value that represents the number of licenses of the specified type that are currently in use.

Exceptions

`GetUsedCount` throws an `ApplicationException` if license information is unavailable because an LDAP error occurred.

Count

Gets the total number of parent license containers in the Active Directory forest.

Syntax

```
int Count {get;}
```

Property value

The number of parent license containers in the collection of parent containers.

HasEvaluation

Indicates whether any generated license key installed in any of the parent license containers is an evaluation license.

Syntax

```
bool HasEvaluation {get;}
```

Property value

Returns `true` if any evaluation license is installed, or `false` if there are no evaluation licenses installed.

HasMachineLicense

Indicates whether any computer license is installed.

Syntax

```
bool HasMachineLicense {get;}
```

Property value

Returns `true` if any computer license is installed, or `false` if there are no computer licenses installed.

Item

Gets the parent license container object by index number.

Syntax

```
ILicenses this[int index] {get;}
```

Parameter

Specify the following parameter when using this property.

index	The index number for retrieving the parent license container object

Return value

The parent container object found at the specified index number.

Map

The Map class contains methods and properties used to manage individual NIS map entries stored in Active Directory. This class is defined in the Centrify.DirectControl.NISMap.API namespace rather than the Centrify.DirectControl.API namespace.

Syntax

```
public class IMap : ICloneable, IDisposable
```

Discussion

The Map class supports the methods and properties that apply to all .NET objects. In addition to those methods and properties, the Map class provides some Delinea-specific methods and properties for managing individual NIS map records. Only the Delinea-specific methods and properties are described in this reference.

Methods

The Map class provides the following Delinea-specific methods:

| Method | Description | | ----- | ----- | | Unexpected Link Text | Adds a new map entry with the specified key to the NIS map. | | Clone | Makes a clone of the NIS map entry. Inherited from ICloneable. | | Unexpected Link Text | Commits changes to the NIS map and saves them in Active Directory. | | 2 Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable. | | Unexpected Link Text | Checks whether there is an entry with a specific key-value pair. | | Unexpected Link Text | Removes the map entry with the specified ID |

Properties

The Map class provides the following Delinea-specific properties:

Unexpected Link Text	Indicates whether the map object is readable.
Unexpected Link Text	Indicates whether the map object is writable.
Unexpected Link Text	Gets or sets the map name of the NIS map.
Unexpected Link Text	Gets the Store instance associated with the map object.
Unexpected Link Text	Gets or sets the NIS map type.

Add

Adds a new map entry, or individual map record, with the specified key.

Syntax

```
Entry Add(string key, string value, string comment);

Entry Add(string key);
```

Parameters

Specify the following parameters when using this method.

key	String	The key field that uniquely defines this entry in the NIS map.
value	String	The value associated with the key field for this entry in the NIS map.
comment	String	Any optional text string to store in the comment field for this entry in the NIS map.

Return value

An entry object instance for the map entry added.

Discussion

Each map entry consists of three primary fields: a key field, a value field, and an optional comment field. The content and format of the value field depends on the type of NIS map you are working with. For example, if you are setting the value field in a generic map, the field can contain virtually any string that you want served for a corresponding key name. If you are adding a map entry to a netgroup, auto.mount, or auto.master map, the single value field defined in Active Directory may consist of multiple, concatenated fields appropriate for the map type.

Because of potential API name conflict, the Add(string key) method can be used with .NET programs only.

Example

The following code sample illustrates using Add to add new NIS entries to different types of NIS maps:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic NIS map named "Remote servers"
Set map = store.open("Remote servers")
'Add an entry to the "Remote servers" NIS map. The input format is:
'map.add "\<key\>", "\<value\>", "\<comment\>"
map.add "mirage", "127.67.10.1", "Server located in Toledo office"
'Open the NIS map named “auto.master”
Set map = store.open("auto.master")
'Add an entry to the "auto.master" NIS map. The input format is: 'map.add
"\<mount point\>", "\<map\>" & Chr(11) & "\<Options\>", "\<comment\>"
'where:
'\<mount point\> - mount point name (key field)
'\<map\> - the map file to consult for this mount point
'\<options\> - mount options
'\<comment\> - text comments
'NOTE: The \<map\> and \<options\> are separated by a tab character, 'Chr(11),
and form the value field for the entry.
map.add "/net", "-hosts" & Chr(11) & "-nosuid,nobrowse", "sample mount"
'Open the NIS map named “auto.mount”
Set map = store.open("auto.mount")
'Add an entry to “auto.mount" NIS map. The input format is:
```

```
'map.add "\<name\>", "\<Options\>" & Chr(11) & "\<network path\>",
""\<comment\>"
'where:
'\<name\> - mount point name (key field)
'\<options\> - mount options
'\<network path\> - the network path to consult for this mount point
'\<comment\> - text comments
'NOTE: The \<options\> and \<network path\> are separated by a tab 'character,
Chr(11), and form the value field for the entry.
map.add "cdrom", "-fstype=nsfs,ro" & Chr(11) & ":/dev/sr0", "sample mount"
'Open the NIS map named “netgroup”
set map = store.open("netgroup")
'Add entries to the “netgroup” NIS map. The input format is:
'map.add "\<group_name\>", "\<member_name\>", "comment"
'where:
'\<group_name\> - defines the unique key of this group
'\<member_name\> - lists the members of the group. Each \<member_name\>
' is either another group name, all of whose members
' are to be included in the group being defined,
' or a triple of the form:
' (hostname,username,domainname)
'\<comment\> - comments of this user
'Add a group "db_admin" with three members: dbas, dean, jon
map.add "db_admin", "dbas (,dean,) (firebird,jon,)", "All DBAs"
wScript.Echo "NIS map entries added."
...
```

Commit

Commits the settings or changes for the map object to Active Directory.

Syntax

```
void Commit();
```

Discussion

The Commit method saves the settings of the map, but not the entries in the map.

Exceptions

Commit throws an ApplicationException if access is denied.

Example

The following code sample illustrates using Commit in a script:

```
SET cims = CreateObject("Centrify.DirectControl.Cims3")
SET zone = cims.GetZone("example.org/Zones/default")
SET store = CreateObject("Centrify.DirectControl.Nis.Store")
store.Attach zone.ADsPath, "username", "password"
SET map = store.Open("computers")
map.Name = "hosts"
map.Commit
```

Exists

Checks whether there is an entry with a specific key-value pair.

Syntax

```
bool Exists(string key, string value)
```

Parameter

Specify the following parameters when using this method:

key	The key you want to check for.
value	The value you want to check for.

Return value

Returns `true` if the specified entry exists.

Get

Returns the map entry, or individual map record, with the specified key.

Syntax

```
Entry Get(string key);
```

Parameter

Specify the following parameter when using this method:

key	The key field that uniquely defines this entry in the NIS map.

Return value

An entry object instance for the NIS map and key specified.

Example

The following code sample illustrates using Get to retrieve an existing NIS map entry from a specific map.

```
...
'Specify the zone you want to work with
set zone = cims.GetZoneByPath("LDAP://CN=qa-slovenia,CN=unix,DC=quantum,DC=net")

'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "nate.james", "my3w0rds"
'Open the NIS map named "Remote servers"
Set objMap = store.open("Remote servers")
'Specify the map entry key field
set objEntry = objMap.Get("helios")
wScript.Echo “IP: “ & objEntry.Value
…
```

GetByID

Returns the map entry, or individual map record, with the specified ID.

Syntax

```
Entry GetByID(string id);
```

Parameter

Specify the following parameter when using this method:

id	The ID that uniquely defines this entry in the NIS map.

Return value

The entry object instance with the specified ID.

GetDirectoryEntry

Returns the directory entry for the NIS map object.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The directory entry for the map object.

Discussion

This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

GetEnumerator

Returns an enumeration of IMap objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of map entries.

GetRedirectMap

Returns the redirect target NIS map.

Syntax

```
Entry GetRedirectMap(Connection connection);
```

Parameter

Specify the following parameter when using this method:

connection	The Cims connection.

Return value

The redirect target NIS map.

Import

Imports a new map entry, or individual map record, with the specified key into Active Directory.

Syntax

```
Entry Import(string key, string value, string comment);
```

Parameters

Specify the following parameters when using this method.

key	The key field that uniquely defines this entry in the NIS map.
value	The value associated with the key field for this entry in the NIS map.
comment	Any optional text string to store in the comment field for this entry in the NIS map.

Return value

An entry object for the map entry imported.

Discussion

The difference between the Import and Add methods is that the Import method performs minimal checking and validation of data to maximize performance.

Exceptions

Import throws an ApplicationException if the key or value is invalid.

Example

The following code sample illustrates using Import to retrieve an existing NIS map entry from a specific map.

```
...
'Specify the zone you want to work with
set zone = cims.GetZoneByPath("LDAP://CN=qa-slovenia,CN=unix,DC=quantum,DC=net")

'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the NIS map named "Remote servers"
Set map = store.open("Remote servers")
'Add an entry to the "Remote servers" NIS map. The input format is:
map.import "oaxaca", "127.67.32.10", "Latin America Support office"
...
```

Remove

Removes the map entry with the specified key.

Syntax

```
void Remove(string key);
```

Parameter

Specify the following parameter when using this method:

key	The key field that uniquely defines this entry in the NIS map.

Exceptions

Remove throws an ApplicationException if it can't find the DirectoryEntry value.

Example

The following code sample illustrates using Remove to remove an existing NIS map entry by key name from a specific map:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and user credentials
store.Attach zone.ADsPath, "jae.smith", "pas\$w0rd"
'Remove the “Modified_Key” map entry from the "generic map" NIS map
set map = store.Open("generic map")
map.Remove "Modified_Key"
wScript.Echo "NIS map entry removed."
...
```

RemoveByID

Removes the map entry with the specified ID.

Syntax

```
void Remove(string id);
```

Parameter

Specify the following parameter when using this method:

id	The ID that uniquely defines this entry in the NIS map.

Exceptions

RemoveByID throws an ApplicationException if it can't find the DirectoryEntry object.

IsReadable

Indicates whether the NIS map in the attached zone is readable.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the map object is readable by the user, or false if the map object is not readable.

Discussion

This property returns a value of true if the user accessing the map entry object in Active Directory has sufficient permissions to read the entry properties.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=offshore,CN=unix,DC=quantum,DC=net")

'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "tae.parker", "9days\^"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Check whether the map is readable
if not map.IsReadable then
wScript.Echo “No read permission. Quitting application ...”
wScript.Quit
end if
…
```

IsWritable

Indicates whether the map object is writable.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns true if the map object is writable by the user, or false if the map object is not writable.

Discussion

This property returns a value of true if the user accessing the map object in Active Directory has sufficient permissions to change the map object's properties.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=pilot,CN=unix,DC=quantum,DC=net")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the generic map type named "Workstations IDs"
Set map = store.open("Workstations IDs")
'Check whether the map is writable
If not map.IsWritable then
wScript.Echo “No write permission for ” & map.Name
wScript.Quit
end if
…
```

Name

Gets or sets the map name.

Syntax

```
string Name {get; set;}
```

Property value

The map name.

Discussion

You can specify any string for this property regardless of the type of NIS map.

Exceptions

`Name` throws an `ArgumentException` if you try to set a value that is `null`, empty, or greater than 64 characters.

Example

The following code sample illustrates using this property in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=pilot,CN=unix,DC=quantum,DC=net")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
objStore.Attach objZone.ADsPath, "jae.smith", "pas\$w0rd"
'List map names
For each map in objStore
wScript.Echo map.Name
end if
...
```

Store

Gets the store object instance for the map object.

Syntax

```
Store Store {get;}
```

Property value

The Store instance for the map object.

Type

Gets or sets the map type for the map object.

Syntax

```
string Type {get; set;}
```

Property value

The map type for the map object.

Discussion

Internally, the map type defines how fields are parsed and interpreted for standard network maps and generic maps. the type value is used by Access Manager to identify the map type. Access Manager can recognize all of the common NIS map types.

Example

The following code sample illustrates using this property in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=pilot,CN=unix,DC=quantum,DC=net")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
objStore.Attach objZone.ADsPath, "jae.smith", "pas\$w0rd"
'Open the map type named "Workstations IDs"
Set objMap = objStore.Open("Workstations IDs")
'Check the map type
wScript.Echo “Map Type: ” & objMap.Type
...
```

Delinea

MzRoleAssignment

Represents a computer-level role assignment.

Syntax

public interface IMzRoleAssignment : IRoleAssignment

Methods

The MzRoleAssignment class provides the following methods:

| Method | Description | | ----- | ----- | | Unexpected Link Text | VBScript interface to clear the custom attributes for this class. (Inherited from Unexpected Link Text.) | | Unexpected Link Text | Commits changes in the role assignment to Active Directory. (Inherited from RoleAssignment.) | | Unexpected Link Text | Deletes the role. (Inherited from Unexpected Link Text.) |

Properties

The MzRoleAssignment class provides the following properties:

Unexpected Link Text	VBScript only: Gets or sets custom attributes for this class.
Unexpected Link Text	Determines the time at which this role becomes inactive. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the GUID of the role assignment. (Inherited from RoleAssignment.)
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid role. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid trustee. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the local trustee being assigned. (Inherited from RoleAssignment.)
Unexpected Link Text	Gets or sets the role the trustee is assigned to. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the time from which this role becomes effective. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the distinguished name of the trustee assigned this role. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the trustee type of the role assignment. (Inherited from Unexpected Link Text.)

GetComputer

Returns the computer for which the role assignment is made.

Syntax

```
IComputer GetComputer()
```

Return value

The computer object representing the computer for which the role assignment is made.

Exceptions

`GetComputer` throws an `ApplicationException` if there are multiple computers, computer service connection points (SCPs), or zones that have the same DNS host name; if the method failed to find the computer; or if the method failed to get the computer profile from the role assignment.

NetworkAccess

This class represents a network access right.

Syntax

```
public interface INetworkAccess:IRight
```

The NetworkAccess class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Removes the right. (Inherited from Unexpected Link Text.)

Properties

The NetworkAccess class provides the following properties:

Unexpected Link Text	Gets or sets the description of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the priority of this right.
Unexpected Link Text	Gets or sets whether the user's password is required when this right is used.
Unexpected Link Text	Gets or sets the SID for the run-as user or an SID for the user assigned the right (VBScript).
Unexpected Link Text	Gets or sets the SID for the run-as user or a list of SIDs for the users assigned the right (.NET).
Unexpected Link Text	Gets or sets the run-as type for the right.
Unexpected Link Text	Gets the zone this right belongs to. (Inherited from Unexpected Link Text.)

Discussion

A network access right enables a user to run an application on a remote computer as another user. For example, a network access right can give a user the ability to run as an SQL Administrator on a remote server.

Priority

Gets or sets the priority of this right.

Syntax

```
int Priority {get; set;}
```

Property value

The priority of the right. Default is 0.

Discussion

This number is used when handling multiple matches for rights specified by wild cards. If rights specified by this property object match rights specified by another property object, the object with the higher priority prevails. The higher the value of the Priority property, the higher the priority.

RequirePassword

Gets or sets whether the logged-in user's password is required when this right is used.

Syntax

```
bool RequirePassword {get; set;}
```

Property value

Set to `true` if the right requires the logged-in user's password.

RunAs

Gets or sets the run-as property for this right.

Syntax

```
string RunAs {get; set;}
```

Property value

The run-as property for a single user.

Discussion

If the Unexpected Link Text property is set to Self, the remote application is run under the logged-in user account, but with the additional privileges of the user whose SID is listed in the RunAs property. For example, if the NetworkAccess right is set to run as Self and RunAs contains the SID of the Network Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Network Admins group.

If the RunAsType property is set to User, the remote application is run under the user whose SID is listed in the RunAs property. For example, if the NetworkAccess right is set to run as User and RunAs contains the SID of the user NetAdmin, then this application runs with the permissions of the NetAdmin user.

If the RunAs property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

> **Note**: This property is for use in VBScript programs. Use the RunAsList property for .NET.

RunAsList

Gets or sets the run-as list for this right.

Syntax

IList\<SecurityIdentifier\> RunAsList {get; set;}

Property value

The run-as list for the right.

Discussion

If the Unexpected Link Text property is set to Self, the remote application is run under the logged-in user account, but with the additional privileges of the groups whose SIDs are listed in the RunAsList property. For example, if the NetworkAccess right is set to run as Self and RunAsList contains the SID of the Network Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Network Admins group.

If the RunAsType property is set to User, the remote application is run under the user whose SID is listed in the RunAsList property. In this case, the RunAsList property contains only a single SID. For example, if the NetworkAccess right is set to run as User and RunAsList contains the SID of the user NetAdmin, then this application runs with the permissions of the NetAdmin user.

If the RunAsList property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

Note: This property can only be used in .NET programs. Use the RunAs property for VBScript.

RunAsType

Gets or sets the run-as type for this right.

Syntax

```
WindowsRunAsType RunAsType {get; set;}
```

Property value

The run-as type of the right.

Possible values:

```
public enum WindowsRunAsType
{
    // Run as self
    Self,
    // Run as another user
    User
}
```

Discussion

If the RunAsType property is set to Self, the remote application runs as the logged-in user with the additional privileges of the groups whose SIDs are listed in the Unexpected Link Text. For example, if the NetworkAccess right is set to run as Self and RunAsList contains the SID of the Network Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Network Admins group.

If the RunAsType property is set to User, the application is run as the user whose SID is listed in the RunAsList property. For example, if the NetworkAccess right is set to run as User and RunAsList contains the SID of the user NetAdmin, then this application runs as NetAdmin with the permissions of that user.

NetworkAccesses

The NetworkAccesses class manages a collection of network access rights.

Syntax

public interface INetworkAccesses

Methods

The NetworkAccesses class provides the following method:

Unexpected Link Text	Gets the enumerator you can use to enumerate all network access rights.

GetEnumerator

Returns an enumeration of NetworkAccess objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all the NetworkAccess objects.

Pam

Represents a PAM application access right.

Syntax

```
public interface IPam: IRight
```

Discussion

A PAM (Pluggable Authentication Module) application right gives a user the ability to access the authorized PAM-enabled application.

Methods

The Pam class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Deletes the right. (Inherited from Unexpected Link Text.)

Properties

The Pam class provides the following properties:

Unexpected Link Text	Gets or sets the PAM application.
Unexpected Link Text	Gets or sets the description of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the zone this right belongs to. (Inherited from Unexpected Link Text.)

Application

Gets or sets the PAM application for which this is a right.

Syntax

```
string Application {get; set;}
```

Property value

The file path of the application.

Exceptions

Application throws an ArgumentException if you try to set the property and the string is null or empty.

Example

The following code sample illustrates using Application in a script:

```
...
string strParent = "CN=zones,CN=Centrify,CN=Program Data";
if (args.Length != 3)
{
   Console.WriteLine("Usage:");
   Console.WriteLine(" test_add_pam.exe \\"zone-name\\" \\"pam-name\\"
      \\"pam-application\\"");
   return;
}
string strZone = args[0];
string strName = args[1];
string strApp = args[2];
// Need to obtain an active directory container object
DirectoryEntry objRootDSE = new DirectoryEntry("LDAP://rootDSE");
DirectoryEntry objContainer = new DirectoryEntry("LDAP://" + strParent + "," +
objRootDSE.Properties["defaultNamingContext"].Value.ToString());
string strContainerDN = objContainer.Properties["DistinguishedName"].Value as
string;
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
}
else
{
   IPam objPam = objZone.GetPamAccess(strName);
   if (objPam != null)
   {
      Console.WriteLine("PAM " + strName + " already exist.");
   }
   else
   {
      objPam = objZone.CreatePamAccess();
      objPam.Name = strName;
      objPam.Application = strApp;
      objPam.Description = "optional description";
      objPam.Commit();
   }
}
...
```

Pams

The Pams class manages a collection of PAM application access rights.

Syntax

```
public interface IPams
```

Methods

The Pams class provides the following method:

Unexpected Link Text	Returns the enumerator you can use to enumerate all PAM rights.

GetEnumerator

Returns an enumeration of PAM access rights.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

An enumerator you can use to list all the Pam objects.

Right

This class provides a base class for all rights.

Syntax

public interface IRight

Methods

The Right class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory.
Unexpected Link Text	Deletes the right.

Properties

The Right class provides the following properties:

Unexpected Link Text	Gets or sets the description of the right.
Unexpected Link Text	Indicates whether the right is readable.
Unexpected Link Text	Indicates whether the right is writable.
Unexpected Link Text	Gets or sets the name of the right.
Unexpected Link Text	Gets the zone this right belongs to.

Commit

Commits any changes or updates to the Right object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

The method does not validate the data before saving it in Active Directory.

Exceptions

Commit throws an ApplicationException if:

- The command right name or command pattern is null, empty, or invalid.
- The command name is null or empty.
- The command path is null or empty.
- The method cannot find the command right or authorization data for the zone.

If the Commit method fails, see the message returned by the exception for more specific information about the reason for the failure.

Delete

Deletes the right from Active Directory.

Syntax

```
void Delete()
```

Exceptions

Delete throws an ApplicationException if the method cannot find the command right to delete or cannot find authorization data for the zone.

If the Delete method fails, see the message returned by the exception for more specific information about the reason for the failure.

Description

Gets or sets the description of the right.

Syntax

```
string Description {get; set;}
```

Property value

A string describing the right.

IsReadable

Indicates whether the right is readable.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns `true` if the right is readable.

IsWritable

Indicates whether the right is writable.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns `true` if the right is writable.

Name

Gets or sets the name of the right.

Syntax

```
string Name {get; set;}
```

Property value

The name of the right. The name can contain only letters (upper- or lowercase), numerals 0 through 9, and the hyphen (-) and underscore (_) characters.

Exceptions

`Name` throws an `ArgumentException` if the name is `null`, empty, or contains invalid characters.

Zone

Indicates the zone to which this right belongs.

Syntax

IZone Zone {get;}

Property value

The zone.

Delinea

Role

The Role class represents a user access role.

Syntax

public interface IRole

Methods

The Role class provides the following methods:

Unexpected Link Text	Adds a command right to the role.
Unexpected Link Text	Adds a network application access right to the role.
Unexpected Link Text	Adds a PAM application access right to the role.
Unexpected Link Text	Adds an SSH application access right to the role.
Unexpected Link Text	Adds a Windows application right to the role.
Unexpected Link Text	Adds a Windows desktop right to the role.
Unexpected Link Text	Adds a trustee to the role at the zone or computer level.
Unexpected Link Text	VBScript interface to clear the custom attributes for this class.
Unexpected Link Text	Commits changes to the role to Active Directory.
Unexpected Link Text	Deletes the role from Active Directory.
Unexpected Link Text	Returns all command rights added to the role.
ICustomAttributeContainer Unexpected Link Text	.NET interface that returns the directory entry for the parent container object for the custom attributes for this class.
Unexpected Link Text	Returns the collection of all network application access rights added to this role.
Unexpected Link Text	Returns all PAM application access rights added to the role.
Unexpected Link Text	Returns all SSH application access rights added to the role.
Unexpected Link Text	Returns the collection of all Windows application rights added to this role.
Unexpected Link Text	Returns the collection of all Windows desktop rights added to this role.
Unexpected Link Text	Indicates whether the role is valid in a specified time period.
Unexpected Link Text	Removes all rights from the role.
Unexpected Link Text	Removes a specific command right from the role.
Unexpected Link Text	Removes a specific PAM application right from the role.
Unexpected Link Text	Removes a specific PAM application right from the role.

Unexpected Link Text	Removes a specific SSH application right from the role.
Unexpected Link Text	Removes a specific Windows application access right from the role.
Unexpected Link Text	Removes a specific Windows desktop right from the role.
Unexpected Link Text	Sets a day of the week on which the role is active or inactive.
Unexpected Link Text	Sets a day and hour of the week for which the role is active or inactive.
Unexpected Link Text	VBScript interface to set the custom attributes for this class.

Properties

The Role class provides the following properties:

Unexpected Link Text	Determines whether the role allows local users.
Unexpected Link Text	Gets or sets the time at which the role is active, in hex format.
Unexpected Link Text	VBScript only: Gets or sets custom attributes for this class.
Unexpected Link Text	Gets or sets the description of the role.
Unexpected Link Text	Gets the GUID for this role.
Unexpected Link Text	Indicates whether the role is readable.
Unexpected Link Text	Indicates whether the role is writable.
Unexpected Link Text	Gets or sets the name of the role.
Unexpected Link Text	Gets or sets the system rights granted to the role.
Unexpected Link Text	Gets the zone to which this role belongs.

AddCommand

Adds a command right to the role.

Syntax

```
void AddCommand(Icommand command)
```

Parameter

Specify the following parameter when using this method:

command	The command right you want to add to the role.

Discussion

This command right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

AddCommand throws an ApplicationException if the command right is not in the current or parent zone.

Example

The following code sample illustrates using AddCommand in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
   Console.WriteLine("Zone " + strZone + " does not exist.");
   return;
}
IRole objRole = objZone.GetRole(strRole);
if (objRole == null)
{
   Console.WriteLine("Role " + strRole + " does not exist.");
   return;
}
ICommand objCmd = objZone.GetCommand(strCmd);
if (objCmd == null)
{
   Console.WriteLine("Command " + strCmd + " does not exist.");
   return;
}
objRole.AddCommand(objCmd);
objRole.Commit();
...
```

AddNetworkAccess

Adds a network application access right to the role.

Syntax

```
void AddNetworkAccess(INetworkAccess networkAccess)
```

Parameter

Specify the following parameter when using this method:

networkAccess	The network application right you want to add to the role.

Discussion

This right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

AddNetworkAccess throws an ApplicationException if the network access right is not in the current or parent zone.

AddPamAccess

Adds a PAM application access right to the role.

Syntax

```
void AddPamAccess(IPam pam)
```

Parameter

Specify the following parameter when using this method:

pam	The PAM application right you want to add to the role.

Discussion

This right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

AddPamAccess throws an ApplicationException if the PAM application access right is not in the current or parent zone.

AddSsh

Adds an SSH application access right to the role.

Syntax

```
void AddSsh(ISsh ssh)
```

Parameter

Specify the following parameter when using this method:

ssh	The SSH application right you want to add to the role.

Discussion

This right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

`AddSsh` throws an `ApplicationException` if the SSH application right is not in the current or parent zone.

AddWindowsApplication

Adds a Windows application right to the role.

Syntax

```
void AddWindowsApplication(IWindowsApplication windowsApplication)
```

Parameter

Specify the following parameter when using this method:

windowsApplication	The Windows application right you want to add to the role.

Discussion

This right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

AddWindowsApplication throws an ApplicationException if the Windows application right is not in the current or parent zone.

AddWindowsDesktop

Adds a Windows desktop right to the role.

Syntax

```
void AddWindowsDesktop(IWindowsDesktop windowsDesktop)
```

Parameter

Specify the following parameter when using this method:

windowsDesktop	The Windows desktop right you want to add to the role.

Discussion

This right is not stored in Active Directory until you call the Unexpected Link Text method.

Exceptions

AddWindowsDesktop throws an ApplicationException if the Windows desktop right is not in the current or parent zone.

Example

The following code sample illustrates using AddWindowsDesktop in a script:

```
...
// Get the zone object
IHierarchicalZone objZone =
cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN) as IHierarchicalZone;

if (objZone == null)
{
    Console.WriteLine("Zone " + strZone + " does not exist.");
    return;
}
IRole objRole = objZone.GetRole(strRole);
if (objRole == null)
{
    Console.WriteLine("Role " + strRole + " does not exist.");
    return;
}
IWindowsDesktop objWindowsDesktop =
objZone.GetWindowsDesktop(strWindowsDesktop);
if (objWindowsDesktop == null)
{
    Console.WriteLine("WindowsDesktop " + strWindowsDesktop + " does not exist.");
    return;
}
objRole.AddWindowsDesktop(objWindowsDesktop);
objRole.Commit();
...
```

Assign

Assigns a trustee to a role at the zone or computer level.

Syntax

```
IRoleAssignment Assign(DirectoryEntry trusteeDE, IComputer computer)
IRoleAssignment Assign(DirectoryEntry trusteeDE, IZone zone)
IRoleAssignment Assign(SearchResult trusteeSR, Icomputer computer)
IRoleAssignment Assign(SearchResult trusteeSR, IZone zone)
IRoleAssignment Assign(string trusteeDN, IComputer computer)
IRoleAssignment Assign(string trusteeDN, IZone zone)
```

Parameters

Use the following parameters with this method.

trusteeDE	The directory entry for the trustee (user or group) you want to add.
trusteeSR	The directory entry for a trustee specified as a search result.
trusteeDn	The trustee specified as a distinguished name.
computer	The computer to which you want to add the role.
zone	The zone to which you want to add the role.

Return value

The role assignment that includes the specified trustee. This role assignment is not stored in Active Directory until you call the Unexpected Link Text method.

Discussion

The `Assign(DirectoryEntry trusteeDE, IComputer computer)`, `Assign(SearchResult trusteeSr, IComputer computer)`, `Assign(DirectoryEntry trusteeDE, IZone zone)` and `Assign(SearchResult trusteeSr, IZone zone)` methods are available only for .NET-based programs.

Exceptions

Assign may throw one of the following exceptions:

- `ApplicationException` if the trustee is not a user or a group, you attempt to assign a role to a zone other than the containing or child zone, the method fails to create a role assignment (see the message returned by the exception for the reason), the method cannot find the trustee object or distinguished name in the specified search result, or the method cannot find the trustee.
- `ArgumentException` if any parameter is `null` or empty.

Commit

Commits any changes or updates to the Role object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

The method does not validate the data before saving it in Active Directory.

Exceptions

`Commit` throws an `ApplicationException` if:

- The role name is `null`, empty, or invalid.
- The method cannot find the role or command right.
- The method cannot find authorization data for the zone.

If the `Commit` method fails, see the message returned by the exception for more specific information about the reason for the failure.

Delete

Deletes the role from Active Directory.

Syntax

```
void Delete()
```

Exceptions

Delete throws an ApplicationException if the method cannot find the role or authorization data for the zone.

If the Delete method fails, see the message returned by the exception for more specific information about the reason for the failure.

GetCommands

Returns all the command rights added to this role.

Syntax

```
ICommands GetCommands()
```

Return value

The collection of command rights. Enumerate this object to get all of the ICommand objects for this role.

GetNetworkAccesses

Returns the collection of all network application access rights added to this role.

Syntax

```
INetworkAccesses GetNetworkAccesses()
```

Return value

A collection of NetworkAccess objects representing all the network application access rights in this role.

GetPamAccesses

Returns the collection of all PAM application rights added to this role.

Syntax

```
IPams GetPamAccesses()
```

Return value

A collection of Unexpected Link Text objects representing all the PAM application rights in this role.

GetSshRights

Returns the collection of all SSH application rights added to this role.

Syntax

```
ISshs GetSshRights()
```

Return value

A collection of Unexpected Link Text objects representing all the SSH application rights in this role.

GetWindowsApplications

Returns the collection of all Windows application rights added to this role.

Syntax

```
IWindowsApplications GetWindowsApplications()
```

Return value

A collection of Unexpected Link Text objects representing all the Windows application rights in this role.

GetWindowsDesktops

Returns the collection of all Windows desktop rights added to this role.

Syntax

```
IWindowsDesktops GetWindowsDesktops()
```

Return value

A collection of Unexpected Link Text objects representing all the Windows desktop rights in this role.

IsApplicable

Checks to see if the role is valid at a specified time.

Syntax

```
bool IsApplicable(int dayOfWeek, int hourOfDay)
```

Parameter

Specify the following parameters when using this method.

dayOfWeek	The day of the week, where 0 is Sunday and 6 is Saturday.
hourOfDay	The hour of the day, where 0 is midnight and 23 is 11:00 PM

Return value

Returns `true` if the role is valid at the specified time.

Discussion

If the role is active at a particular hour, it is active for that entire hour. To set a specific hour, see Unexpected Link Text. To set an entire day, see Unexpected Link Text. To set up an entire schedule with one call, see Unexpected Link Text.

RemoveAllRights

Removes all rights from the role.

Syntax

```
void RemoveAllRights()
```

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemoveCommand

Removes a specific command right from the role.

Syntax

```
void RemoveCommand(ICommand command)
```

Parameter

Specify the following parameter when using this method:

command	The command right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemoveNetworkAccess

Removes a specific network access right from the role.

Syntax

```
void RemoveNetworkAccess(INetworkAccess networkAccess)
```

Parameter

Specify the following parameter when using this method:

networkAccess	The network application access right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemovePamAccess

Removes a specific PAM application access right from the role.

Syntax

```
void RemovePamAccess(IPam pam)
```

Parameter

Specify the following parameter when using this method:

pam	The PAM application access right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemoveSshRight

Removes a specific SSH application access right from the role.

Syntax

```
void RemoveSshRight(ISsh ssh)
```

Parameter

Specify the following parameter when using this method:

ssh	The SSH application access right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemoveWindowsApplication

Removes a specific Windows application access right from the role.

Syntax

```
void RemoveWindowsApplication(IWindowsApplication windowsApplication)
```

Parameter

Specify the following parameter when using this method:

windowsApplication	The Windows application access right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

RemoveWindowsDesktop

Removes a specific Windows desktop right from the role.

Syntax

```
void RemoveWindowsDesktop(IWindowsDesktop windowsDesktop)
```

Parameter

Specify the following parameter when using this method:

windowsDesktop	The Windows desktop right you want to remove.

Discussion

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

SetApplicableDay

Sets a day of the week on which the role is active or inactive.

Syntax

```
void SetApplicableDay(int dayOfWeek, bool isApplicable)
```

Parameter

Specify the following parameters when using this method.

dayOfWeek	The day of the week on which you want the role to be active or inactive, where 0 is Sunday and 6 is Saturday.
isApplicable	Set true to make the role active on the specified day, or false to make the role inactive on that day.

Discussion

If you set the role active on Monday, it is active all day each Monday. To set a specific hour, see Unexpected Link Text. To set up an entire schedule with one call, see Unexpected Link Text. To check whether the role is active at a given time, see Unexpected Link Text.

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

SetApplicableHour

Sets a day and hour of the week for which the role is active or inactive.

Syntax

```
void SetApplicableHour(int dayOfWeek, int hourOfDay, bool isApplicable)
```

Parameter

Specify the following parameters when using this method.

dayOfWeek	The day of the week on which you want the role to be active or inactive, where 0 is Sunday and 6 is Saturday.
hourOfDay	The hour of the day at which you want the role to be active or inactive, where 0 is midnight and 23 is 11:00 PM
isApplicable	Set true to make the role active on the specified day and hour, or false to make the role inactive.

Discussion

If you set the role active on Monday at 10 AM, it is active every Monday from 10:00 to 10:59. To set an entire day, see Unexpected Link Text. To set up an entire schedule with one call, see Unexpected Link Text. To check whether the role is active at a given time, see Unexpected Link Text.

Changes to the role assignments are not stored in Active Directory until you call the Unexpected Link Text method.

AllowLocalUser

Determines whether the role allows local users.

Syntax

```
bool AllowLocalUser {get; set;}
```

Property value

Set to `true` if the role allows local users. The default is `false`.

ApplicableTimeHexString

Gets or sets the time at which the role is active, specified as a hexadecimal number.

Syntax

```
string ApplicableTimeHexString {get; set;}
```

Property value

The times at which the role is active or inactive.

Discussion

This is a 42-character (21-byte) hexadecimal value stored as a string. When the hex value is converted to a binary value, its 168 bits each map to a single hour within the week. If a bit is set to 1, its corresponding hour is enabled for the role. If set to 0, its corresponding hour is disabled.

For details of how the bits are mapped to the hours of the week, see Reading and setting timebox values

To set a specific hour, see Unexpected Link Text. To set an entire day, see Unexpected Link Text. To check whether the role is active at a given time, see Unexpected Link Text.

Exceptions

`ApplicableTimeHexString` throws an `ArgumentException` if the hex string is invalid.

Description

Gets or sets a description of the role.

Syntax

```
string Description {get; set;}
```

Property value

A description of the role.

Guid

Gets the GUID for this role.

Syntax

```
Guid Guid {get;}
```

Property value

The GUID. If the role has not been saved, this property returns Guid.Empty.

IsReadable

Indicates whether the role is readable.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns `true` if the role is readable.

Discussion

This property returns a value of true if the user accessing the `Role` object in Active Directory has sufficient permissions to read its properties.

IsWritable

Indicates whether the role is writable.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns `true` if the role is writable.

Discussion

This property returns a value of true if the user accessing the `Role` object in Active Directory has sufficient permissions to write its properties.

Name

Gets or sets the name of the role.

Syntax

```
string Name {get; set;}
```

Property value

The name of the role. The name can contain only letters (upper- or lowercase), numerals 0 through 9, and the hyphen (-) and underscore (_) characters.

Exceptions

`Name` throws an `ArgumentException` if the name is `null` or empty or contains invalid characters.

SystemRights

Gets or sets system rights granted to the role.

Syntax

```
SystemRight SystemRights {get; set;}
```

Property value

A byte indicating which system rights are granted.

Possible values:

```
public enum SystemRight
{
    // No system rights
       None = 0,
    // Log in with password
       LoginWithPassword = 1,
    // Log in without password (single sign-on)
       LoginWithoutPassword = 2,
    // Ignore disabled status in Active Directory and log in anyway
       IgnoreDisabled = 4,
    // Allow using a full shell
       AllowNonRestrictedShell = 8,
    // NoAudit
       NoAudit = 16,
    // Audit always required
       AuditRequired = 32
    // Multi-factor authentication required
       MfaRequired = 512,
    // Permit login when running in emergency mode
       Rescue = 64
    // Allow logging in from the console
       ConsoleLogon = 128
    // Allow logging in remotely (RDP)
       RemoteLogon = 256
    // Allow powershell remote access
       PsRemote = 1024
}
```

Discussion

The Rescue system right allows the user to log in when there are problems with the authorization cache or the auditing service that are preventing all other users from logging in. For example, if auditing is required but the auditing service is not running or not available, only users with the rescue system right will be allowed to log in. The rescue system right requires the Delinea NSS module to be running in “emergency” mode because the adclient process is not running.

Zone

Gets the zone to which this role belongs.

Syntax

```
IZone Zone {get;}
```

Property value

The zone.

Delinea

RoleAssignment

This class represents a zone-level role assignment.

Syntax

```
public interface IRoleAssignment
```

Methods

The RoleAssignment class provides the following methods:

Unexpected Link Text	VBScript interface to clear the custom attributes for this class.
Unexpected Link Text	Commits changes in the role assignment to Active Directory.
Unexpected Link Text	Deletes the role assignment.
ICustomAttributeContainer Unexpected Link Text	.NET interface that returns the directory entry for the parent container object for the custom attributes for this class.
Unexpected Link Text	Returns the trustee being assigned.
Unexpected Link Text	VBScript interface to set the custom attributes for this class.
Unexpected Link Text	Validates the role assignment

Properties

The RoleAssignment class provides the following properties:

Unexpected Link Text	VBScript only: Gets or sets custom attributes for this class.
Unexpected Link Text	Determines the time at which this role assignment becomes inactive.
Unexpected Link Text	Gets the GUID of the role assignment.
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid role.
Unexpected Link Text	Indicates whether the role assignment is orphaned due to a missing or invalid trustee.
Unexpected Link Text	Gets or sets the local trustee being assigned.
Unexpected Link Text	Gets or sets the role the trustee is assigned to.
Unexpected Link Text	Gets or sets the time from which this role assignment becomes effective.
Unexpected Link Text	Gets or sets the distinguished name of the trustee assigned this role.
`TrusteeType	Gets or sets the trustee type of the role assignment.

Commit

Commits any changes or updates to the role assignment and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

The method does not validate the data before saving it in Active Directory. Call the Unexpected Link Text method before calling the Commit method to make sure the data is valid.

Exceptions

Commit throws an ApplicationException if:

- The role assignment already exists.
- The method cannot find the role assignment or the role.
- The method cannot find authorization data for the zone.

If the Commit method fails, see the message returned by the exception for more specific information about the reason for the failure.

Delete

Deletes the role assignment from Active Directory.

Syntax

```
void Delete()
```

Exceptions

`Delete` throws an `ApplicationException` if the method cannot find the role assignment or authorization data for the zone.

If the `Delete` method fails, see the message returned by the exception for more specific information about the reason for the failure.

GetTrustee

Returns the trustee assigned to the role.

Syntax

```
DirectoryEntry GetTrustee()
```

Return value

The directory entree of the user or group assigned by this role assignment.

Exceptions

`GetTrustee` throws an `ApplicationException` if the method fails to get the trustee object from directory services.

Validate

Validates the data in the role assignment object before any changes are committed to Active Directory.

Syntax

```
void Validate()
```

Exceptions

Validate throws an ApplicationException if:

- The role assignment already exists.
- The role is null.
- The method cannot find the role assignment or the role.
- The method cannot find authorization data for the zone.
- The start time is later than the end time.

If the Validate method fails, see the message returned by the exception for more specific information about the reason for the failure.

EndTime

Gets or sets the time after which this role assignment is inactive.

Syntax

```
DateTime EndTime {get; set;}
```

Property value

The time at which the role assignment ceases to be active. A value of `DateTime.MaxValue` means the role assignment never expires. The time must be later than 1/1/1970 00:00.

Exceptions

`EndTime` throws an `ArgumentException` if you try to set the end time earlier than the start time or earlier than 2400 hours UTC, 1 January, 1970.

Id

Gets the GUID of the role assignment.

Syntax

```
Guid Id {get;}
```

Property value

The GUID of the role assignment.

IsRoleOrphaned

Indicates whether the role assignment is orphaned due to a missing or invalid role.

Syntax

```
bool IsRoleOrphaned {get;}
```

Property value

Returns `true` if this role assignment is an orphan.

IsTrusteeOrphaned

Indicates whether the role assignment is orphaned due to a missing or invalid user or group.

Syntax

```
bool IsTrusteeOrphaned {get;}
```

Property value

Returns `true` if this role assignment is an orphan.

LocalTrustee

Gets or sets the local user or group being assigned.

Syntax

```
string LocalTrustee {get; set;}
```

Property value

The local trustee; either a group or user. You can set either a name or ID for a local user. A local group string begins with the type flag %. You cannot specify a null or empty string. A user or group name string must match the regular expression (Regex):

@"^%?[\.a-zA-Z0-9_-]+\$?@localhost$"

A user ID string must match the regular expression:

@"^#[0-9]+@localhost$"

Exceptions

LocalTrustee may throw one of the following exceptions:

- ArgumentException if the local trustee string is null or empty.
- ApplicationException if the method fails to update the role assignment (see the message returned by the exception for the reason).

Role

Gets or sets the role.

Syntax

```
IRole Role {get; set;}
```

Property value

The object representing the role.

Exceptions

`Role` throws an `ApplicationException` if the role you specify is not in the current or parent zone.

If the `Role` property fails, see the message returned by the exception for more specific information about the reason for the failure.

StartTime

Gets or sets the time from which this role assignment becomes effective.

Syntax

```
DateTime StartTime {get; set;}
```

Property value

The time at which the role assignment becomes active. A value of `DateTime.MinValue` means the role becomes effective immediately. The time must be later than 1/1/1970 00:00.

Exceptions

`StartTime` throws an `ArgumentException` if you try to set the start time later than the end time or earlier than 2400 hours UTC, 1 January, 1970.

TrusteeDn

Gets or sets the user associated with the role.

Syntax

```
string TrusteeDn {get; set;}
```

Property value

The distinguished name of the user.

Exceptions

`TrusteeDn` may throw one of the following exceptions:

- `ArgumentException` if the trustee string is null or empty.
- `ApplicationException` if the method fails to update the role assignment trustee (see the message returned by the exception for the reason).

TrusteeType

Gets or sets the trustee type of the role assignment.

Syntax

```
TrusteeType TrusteeType {get; set;}
```

Property value

The type of trustee.

Possible values:

```
public enum TrusteeType
{
    // Unknown
       Unknown = 0,
    // AD User
       User = 1,
    // AD group
       Group = 2,
    // Local UNIX user
       LocalUser = 4,
    // Local UNIX group
       LocalGroup = 8,
    // All AD Users
       AllADUsers = 16,
    // All local UNIX accounts
       AllUnixUser = 32
    // Local Windows user
       LocalWindowsUser = 64
    // Local Windows group
       LocalWindowsGroup = 128
    // All local Windows users
       AllWindowsUsers = 256
};
```

Exceptions

`TrusteeType` throws an `ArgumentException` if you try to set the trustee type to any value other than `AllADUsers` or `AllUnixUser`.

RoleAssignments

The RoleAssignments class manages a collection of Role Assignment objects.

Syntax

```
public interface IRoleAssignments
```

Methods

The RoleAssignments class provides the following method:

Unexpected Link Text	Returns an enumeration of role assignments.

GetEnumerator

Returns an enumeration of role assignments.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of RoleAssignment objects.

Exceptions

GetEnumerator throws an ApplicationException if it cannot find the scope in the zone, cannot find the role, or cannot find authorization data for the zone.

If the GetEnumerator method fails, see the message returned by the exception for more specific information about the reason for the failure.

Delinea

Roles

The Roles class manages a collection of Role objects.

Syntax

```
public interface IRoles
```

Methods

The Roles class provides the following method:

Unexpected Link Text	Returns an enumeration of roles.

GetEnumerator

Returns an enumeration of roles.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of Role objects.

Exceptions

GetEnumerator throws an ApplicationException if it cannot find authorization data for the zone.

If the GetEnumerator method fails, see the message returned by the exception for more specific information about the reason for the failure.

Ssh

Represents an SSH application access right.

Syntax

```
public interface ISsh: IRight
```

Discussion

An SSH (Secure Shell) application right gives a user the ability to access the authorized SSH-enabled application.

Methods

The Ssh class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory. (Inherited from Right
Unexpected Link Text	Deletes the right. (Inherited from Right.)

Properties

The Ssh class provides the following properties:

Unexpected Link Text	Gets or sets the SSH application.
Unexpected Link Text	Gets or sets the description of the right. (Inherited from Right.)
Unexpected Link Text	Indicates whether the right is readable. (Inherited from Right.)
Unexpected Link Text	Indicates whether the right is writable. (Inherited from Right.)
Unexpected Link Text	Gets or sets the name of the right. (Inherited from Right.)
Unexpected Link Text	Gets the zone this right belongs to. (Inherited from Right.)

Application

Gets or sets the SSH application for which this is a right.

Syntax

```
string Application {get; set;}
```

Property value

The file path of the application.

Exceptions

`Application` throws an `ArgumentException` if the application string is `null` or empty.

Sshs

The Sshs class manages a collection of SSH application access rights.

Syntax

```
public interface ISshs
```

Methods

The Sshs class provides the following method:

Unexpected Link Text	Returns the enumerator you can use to enumerate all SSH rights.

GetEnumerator

Returns an enumeration of SSH access rights.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

An enumerator you can use to list all the Ssh objects.

Store

The Store class contains methods and properties used to manage a zone's NIS maps stored in Active Directory. This class is defined in the Centrify.DirectControl.NISMap.API namespace rather than the Centrify.DirectControl.API namespace.

Syntax

```
public class IStore : ICloneable, IDisposable
```

Discussion

The Store class supports the methods and properties that apply to all .NET objects. In addition to those methods and properties, the Store class provides some Delinea-specific methods and properties for managing network or generic NIS maps in a zone. Only Delinea-specific methods and properties are described in this reference.

Methods

The Store class provides the following Delinea-specific methods:

Unexpected Link Text	Links the Store object to the specified zone.
Unexpected Link Text	Makes a copy of the current Store object instance. Inherited from ICloneable.
Unexpected Link Text	Creates a new NIS map object in the zone in Active Directory.
Unexpected Link Text	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
Unexpected Link Text	Removes the specified NIS map from the zone and Active Directory.
Unexpected Link Text	Checks whether there is a NIS map with the specified name.
Unexpected Link Text	Gets the directory entry for the NIS map container.
Unexpected Link Text	Opens the NIS map with the specified name.

Properties

The Store class provides the following Delinea-specific properties:

Unexpected Link Text	Indicates whether the NIS map store is readable.
Unexpected Link Text	Indicates whether the NIS map store is writable.

Attach

Attaches the Centrify.DirectControl.NisMap.API namespace storage object to the specified zone.

Syntax

```
void Attach(string zonePath, string username, string password)

void Attach(DirectoryEntry zoneEntry)
```

Parameters

Specify the following parameters when using this method.

zonePath	The LDAP path to the zone to which you want to attach the NIS map storage object.
username	The user name to use when linking the NIS map storage object to the specified zone.
password	The user password to use when linking the NIS map storage object to the specified zone.
zoneEntry	The directory entry for the zone to which you want to attach the NIS map storage object. (.NET only)

Exceptions

Attach may throw the following exception:

- COMException if an error occurs in a call to the underlying interface.
- ApplicationException if it fails to locate the NIS map store, the domain controller is read-only, access is not authorized, or an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using Attach to create a map storage object and attach it to a zone:

```
...
'Identify the zone in which the NIS maps will be created
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas$w0rd"
'Use store.Create to add a generic NIS map in this zone.
store.Create "generic map","Generic Map"
'Use store.Create to also add two auto_master NIS maps
store.Create "auto.master","AutoMaster Map"
store.Create "auto_master","AutoMaster Map"
'Use store.Create to add a automount NIS map
store.Create "auto.mount","Automount Map"
'Use store.Create to add a netgroup NIS map
store.Create "netgroup","Netgroup Map"
wScript.Echo "NIS Maps added to " & zone.Name
...
```

Create

Creates a new NIS map with the specified name.

Syntax

```
Map Create(string mapName, string type)
```

Parameters

Specify the following parameters when using this method.

mapName	The name of the NIS map you want to create.
type	The type of NIS map to create.

Return value

The map object created.

Discussion

When you use this method to create NIS maps, you can specify just the map name or the map name and map type. Internally, however, the map name and type defines how fields are parsed and interpreted for standard network maps and generic maps. For example, the netgroup map name can only be used to create a netgroup type of NIS map. In most cases, you should only create generic maps (key value pairs) using this method to prevent NIS maps from becoming unusable due to unexpected formatting.

Exceptions

Create may throw one of the following exceptions:

- COMException if there is an LDAP error. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- ArgumentException if the map name is not valid.

Example

The following code sample illustrates using Create to add new NIS maps to a zone:

```
...
'Identify the zone in which the NIS maps will be created
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas$w0rd"
'Use store.Create to add a generic NIS map in this zone.
store.Create "Contact List","Generic Map"
'Use store.Create to also add the auto_master NIS maps
store.Create "auto_master","AutoMaster Map"
'Use store.Create to add a automount NIS map
store.Create "automounts","Automount Map"
'Use store.Create to add a netgroup NIS map
store.Create "netgroup","Netgroup Map"
wScript.Echo "NIS Maps added to " & zone.Name
```

Delinea

...

Delete

Deletes the specified NIS map.

Syntax

```
void Delete(string mapName)
void Delete(IMap map)
```

Parameter

Specify the following parameter when using this method:

mapName	The name of the NIS map you want to remove.
map	The NIS map you want to remove. (.NET only)

Exceptions

Delete throws a COMException if there is an LDAP error. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using Delete to remove NIS maps from a zone:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas$w0rd"
'Use store.Delete to delete the generic map named "generic map"
store.Delete "generic map"
wScript.Echo "NIS map deleted."
...
```

Exists

Checks whether there is a NIS map with the specified name.

Syntax

```
bool Exists(string mapName)
```

Parameter

Specify the following parameter when using this method:

mapName	The map name whose existence you want to check for.

Return value

Returns true if the specified NIS map exists.

Exceptions

Exists throws a COMException if an error occurs during a call to the underlying interface.

Example

The following code sample illustrates the use of Store.Exists:

```
SET cims = CreateObject("Centrify.DirectControl.Cims3")
SET zone = cims.GetZone("example.org/Zones/default")
SET store = CreateObject("Centrify.DirectControl.Nis.Store")
store.Attach zone.ADsPath, "username", "password"
IF NOT store.Exists("netgroup") THEN
store.Create "netgroup", "408EE104-1864-41fa-B346-19FED4092E68"
END IF
```

GetDirectoryEntry

Returns the directory entry for the NIS map container.

Syntax

```
DirectoryEntry GetDirectoryEntry ()
```

Return value

The DirectoryEntry attribute of the NIS map container.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Open

Opens the NIS map with the specified name.

Syntax

```
Map Open(string mapName);
```

Parameter

Specify the following parameter when using this method:

mapName	The name of the NIS map you want to remove.

Return value

The `map` object instance of the specified map, or `null` if the specified map is not found.

Exceptions

`Open` throws a `COMException` if there is an LDAP error. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using `Open` to access a specific NIS map:

```
...
'Identify the zone you want to work with
set zone = cims.GetZone("sample.com/centrify/zones/default")
'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone.
'Provide the path to the zone and user credentials (username and 'password).
store.Attach zone.ADsPath, "jae.smith", "pas$w0rd"
'Open the NIS map named "Remote servers"
Set map = store.Open("Remote servers")
...
```

IsReadable

Indicates whether the NIS map storage object is readable.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns `true` if the map storage object is readable by the user, or `false` if the map storage object is not readable.

Discussion

This property returns a value of true if the user accessing the NIS map storage object in Active Directory has sufficient permissions to read its properties.

Exceptions

`IsReadable` may throw one of the following exceptions:

- `ApplicationException` if the store entry cannot be located.
- `COMException` if there is an error in a call to the underlying interface.

Example

The following code sample illustrates using this property in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=offshore,CN=unix,DC=quantum,DC=net")

'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "tae.parker", "9days^"
'Check whether the map node is readable
if not store.IsReadable then
wScript.Echo "No read permission. Quitting application ..."
wScript.Quit
end if
...
```

IsWritable

Indicates whether the NIS map storage object is writable.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns true if the map storage object is writable by the user, or false if the map storage object is not writable.

Discussion

This property returns a value of true if the user accessing the NIS map storage object in Active Directory has sufficient permissions to change the storage object's properties.

Exceptions

IsWritable throws a COMException if there is an LDAP error. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this property in a script:

```
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=offshore,CN=unix,DC=quantum,DC=net")

'Create the Store object
Set store = CreateObject("Centrify.DirectControl.Nis.Store")
'Attach to the target zone
'Provide the path to the zone and username and password.
store.Attach objZone.ADsPath, "tae.parker", "9days^"
'Check whether the map node is writable
if not store.IsWritable then
wScript.Echo "No write permission. Quitting application ..."
wScript.Quit
end if
...
```

User

The User class enables Delinea to associate existing Active Directory user accounts with UNIX profiles that contain the attributes required for users to log on to UNIX computers.

Syntax

public interface IUser

Discussion

These additional UNIX-specific attributes that make up the UNIX profile for an Active Directory user are stored and managed within the UserUnixProfile object.

Methods

The User class provides the following methods:

AddUnixProfile	Adds a new UNIX profile for a user to the specified zone.
Commit	Commits the changes to the User object and saves them in Active Directory.
CommitWithoutCheck	Commits the changes to the User object without validating any of the data before saving.
GetDirectoryEntry	Returns an instance of the DirectoryEntry for the user from Active Directory.
GetRoleAssignmentsFromDomain	Returns the collection of all role assignments for a user in a specified domain.
GetRoleAssignmentsFromForest	Returns the collection of all role assignments for a user in a specified forest.
Refresh	Reloads the data in the cache from Active Directory.

Properties

The User class provides the following properties:

AdsiInterface	Gets the ADSI interface of the user object in Active Directory.
ADsPath	Gets the LDAP path to the Active Directory user object.
ID	Gets the UID for the user as a string.
UnixProfiles	Gets the collection of UNIX profiles for the user.

AddUnixProfile

Adds a new UNIX profile for an existing Active Directory user account to the specified zone.

Syntax

IUserUnixProfile AddUnixProfile (IZone zone, int uid, string name, string shell, string homeDir, int primaryGroup)

IUserUnixProfile AddUnixProfile (IZone zone, long uid, string name, string shell, string homeDir, long primaryGroup)

Parameters

Specify the following parameters when using this method.

zone	The destination zone for the new user information.
uid	The UID of the user associated with the profile.
name	The UNIX login name of the user associated with the profile.
shell	The default shell of the user associated with the profile.
homeDir	The default home directory of the user associated with the profile.
primaryGroup	The GID of the primary group of the user associated with the profile.

Return value

A new UserUnixProfile object.

Discussion

There are two versions of this method: one designed for COM-based programs that supports a 32-bit signed number for the uid and primaryGroup arguments and one designed for .NET-based programs that allows a 64-bit signed number for the arguments.

Exceptions

AddUnixProfile throws a NotSupportedException if the zone schema is not supported.

Example

The following code sample illustrates using AddUnixProfile in a script:

```
...
// Create a CIMS object to interact with AD
ICims cims = new Cims();
// Note: There is no cims.connect function.
// By default, this application will use the connection to the domain controller

// and existing credentials from the computer already logged in.
// Get the user object
IUser objUser = cims.GetUserByPath(strUser);
// Get the zone object
IZone objZone = cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN);
IUserUnixProfile objUserUnixProfile;
if (objUser.UnixProfiles.Find(objZone) == null)
{
    // New user for the zone
    long lngUID = objZone.NextUID;
    string strShell = objZone.DefaultShell;
    string strHome = objZone.DefaultHomeDirectory;
    if (bool.Parse(bPrivate))
```

```
    {
        // Create the user as a member of a private group
        objUserUnixProfile = objUser.AddUnixProfile(objZone, lngUID, strUnixAccount,
        strShell, strHome, lngUID);
    }
    else
    {
        // Create the user as a member of the default group
        IGroupUnixProfile objDefaultGroup = objZone.DefaultGroup;
        long lngGID = 10000; // use 10000 as default
        if (objDefaultGroup != null)
        {
            lngGID = objDefaultGroup.GroupId;
        }
        objUserUnixProfile = objUser.AddUnixProfile(objZone, lngUID, strUnixAccount,
            strShell, strHome, lngGID);
    }
    // Enable the UNIX profile for the end user
    objUserUnixProfile.UnixEnabled = true;
    objUserUnixProfile.Commit();
}
else
{
    Console.WriteLine(strUser + " is already a member of " + strZone);
    return;
}
Console.WriteLine("User " + strUser + " was successfully added to zone " +
    strZone + ".");
...
```

Commit

Commits any changes or updates to the User object and saves the changes to Active Directory.

Syntax

```
void Commit()
```

Discussion

When you use this method, it checks and validates the data before saving it in Active Directory.

Exceptions

Commit may throw one of the following exceptions:

- ApplicationException if any field in the UNIX profile is not valid.
- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- UnauthorizedAccessException if you have insufficient access rights to commit changes to the Active Directory object.

Example

The following code sample illustrates using Commit in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the user object
set objUser = cims.GetUserByPath("LDAP://CN=pat.hu,CN=Users, DC=ajax,DC=org")
'Add the UNIX profile for the user
set objUserUnixProfile = objUser.AddUnixProfile(objZone, 623, "pat_hu",
"/bin/bash", "/home/pat_hu", 623)
'Enable the user's UNIX profile
objUserUnixProfile.UnixEnabled = True
'Update Active Directory
objUserUnixProfile.commit
...
```

CommitWithoutCheck

Commits any changes or updates to the User object and saves the changes to Active Directory.

Syntax

```
void CommitWithoutCheck()
```

Discussion

When you use this method, it does not validate any of the data before saving.

Exceptions

CommitWithoutCheck may throw one of the following exceptions:

- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.
- UnauthorizedAccessException if you have insufficient access rights to commit changes on the Active Directory object.

Example

The following code sample illustrates using CommitWithoutCheck in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the user object
set objUser = cims.GetUserByPath("LDAP://CN=pat.hu,CN=Users, DC=ajax,DC=org")
'Add the UNIX profile for the user
set objUserUnixProfile = objUser.AddUnixProfile(objZone, nextuid, unixlogin,
defaultshell, homedir, admingid)
'Enable the user's UNIX profile
objUserUnixProfile.UnixEnabled = True
'Update Active Directory without validating the UNIX profile
objUserUnixProfile.CommitWithoutCheck
...
```

GetDirectoryEntry

Returns the directory entry for the user from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry()
```

Return value

A directory entry for the service connection point that represents the user's UNIX profile.

Discussion

The `DirectoryEntry` object represents the directory object for the user and its associated attributes.

> **Note**: This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

`GetDirectoryEntry` throws an `ApplicationException` if it cannot get the directory object.

GetRoleAssignmentsFromDomain

Returns the collection of all role assignments associated with a user in a specified domain.

Syntax

```
IRoleAssignments GetRoleAssignmentsFromDomain(string domain)
```

Parameters

Specify the following parameter when using this method:

domain	The domain to search for the user's role assignments.

Return value

A collection of role assignment objects representing all of the role assignments explicitly assigned to this user in the specified domain or in the currently joined domain.

Discussion

This method only returns the role assignments that have been explicitly assigned to the user. The method will look for stored credentials to access the specified domain. If there are no stored credentials, the method uses the default credentials for the current user.

If you don't specify a domain by passing an empty string ("") to the method, the method returns role assignments from the currently joined domain.

Example

The following code sample illustrates using GetRoleAssignmentsFromDomain in a script:

```
...
\# New Cims object
\$cims = New-Object ("Centrify.DirectControl.API.Cims");
\# Get IUser object
\$objUserDn = "CN=user1,CN=Users,DC=domain,DC=com";
\$objUser = \$cims.GetUser(\$objUserDn);
\# Get role assignments from domain
\$objUser.GetRoleAssignmentsFromDomain("domain.com")
...
```

GetRoleAssignmentsFromForest

Returns the collection of all role assignments associated with a user in a specified Active Directory forest.

Syntax

```
IRoleAssignments GetRoleAssignmentsFromForest(string forest)
```

Parameters

Specify the following parameter when using this method:

forest	The forest to search for the user's role assignments.

Return value

A collection of role assignment objects representing all of the role assignments explicitly assigned to this user in the specified forest or in the currently joined forest.

Discussion

This method only returns the role assignments that have been explicitly assigned to the user. The method will look for stored credentials to access the specified forest. If there are no stored credentials, the method uses the default credentials for the current user.

If you don't specify a forest by passing an empty string ("") to the method, the method returns role assignments from the currently joined forest.

Example

The following code sample illustrates using GetRoleAssignmentsFromForest in a script:

```
...
\# New Cims object
\$cims = New-Object ("Centrify.DirectControl.API.Cims");
\# Get IUser object
\$objUserDn = "CN=user1,CN=Users,DC=domain,DC=com";
\$objUser = \$cims.GetUser(\$objUserDn);
\# Get role assignments from forest
\$objUser.GetRoleAssignmentsFromForest("forest.com")
...
```

Refresh

Reloads the User object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the user information in the cached object to ensure it is synchronized with the latest information in Active Directory.

Exceptions

Refresh throws a COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using Refresh in a script:

```
...
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=corporate,CN=zones,CN=centrify,CN=program
data,DC=sierra,DC=com")
'Get the user object
set objUser = cims.GetUserByPath("LDAP://CN=pat.hu,CN=Users,DC=ajax,DC=org")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Enable the user's UNIX profile
profile.UnixEnabled = True
'Reload the user object
objUser.Refresh
...
```

AdsiInterface

Gets the ADSI interface of the user object in Active Directory.

Syntax

IADsUser AdsiInterface {get;}

Property value

The ADSI interface of the user object in Active Directory.

Example

The following code sample illustrates using AdsiInterface in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Display the ADSI interfce for the user
wScript.Echo “ADSI: “ & profile.AdsiInterface
...
```

ADsPath

Gets the LDAP path to the Active Directory user object.

Syntax

```
string ADsPath {get;}
```

Property value

The LDAP path to the Active Directory user object.

Discussion

The basic format for the LDAP path is:

```
LDAP://[\<domain\>/]\<attr\>=\<name\>...,dc=\<domain part\>...
```

For example, if the user object is `john.doe` in the organizational unit consultants and the domain is `acme.com`, the LDAP path to the object looks like this:

```
LDAP://cn=john.doe,ou=consultants,dc=acme,dc=com
```

Example

The following code sample illustrates using `ADsPath` in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Display the LDAP path for the user
wScript.Echo “LDAP Path: “ & profile.ADsPath
...
```

ID

Gets the unique identifier for the user as a string value.

Syntax

```
string ID {get;}
```

Property value

The unique identifier for the user as a string.

Example

The following code sample illustrates using ID in a script:

```
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Display the UID for the user
wScript.Echo “User Identifier (UID): “ & profile.ID
…
```

UnixProfiles

Gets all of the UNIX profiles for a specified Active Directory user in the current domain.

Syntax

IUserUnixProfiles UnixProfiles {get;}

Property value

The collection of UNIX profiles for the user.

Discussion

The resulting object, UserUnixProfiles, is the collection of UNIX profiles that have been defined for the user across all zones.

Example

The following code sample illustrates using UnixProfiles in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the user object
set objUser = cims.GetUserByPath("LDAP://CN=tai.wu,CN=Users, DC=ajax,DC=org")
'Look up the user's UNIX profile in the zone
dim objUserUnixProfiles
set objUserUnixProfiles = objUser.UnixProfiles
set objUserUnixProfile = objUserUnixProfiles.Find(objZone)
```

UserInfo

The UserInfo class contains methods and properties used to import and map UNIX user profiles to Active Directory user accounts. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

public interface IUserInfo : IDisposable

Methods

The UserInfo class provides the following methods:

Commit	Commits any changes to the pending import user object and saves them in Active Directory.
Delete	Marks the pending user profile object for deletion from Active Directory.
Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
GetCandidate	Returns the directory object of a user pending import.
Import	Links the pending import user profile with the specified Active Directory user account.
SetCandidate	Sets the directory object of a user pending import.
UpdateStatus	Checks the Active Directory forest for matching or conflicting information that will allow or prevent a pending import user being imported.

Properties

The UserInfo class provides the following properties:

CandidateDN	Gets the distinguished name (DN) of the import candidate.
Gecos	Gets or sets the GECOS field of the UNIX profiles for the pending import user.
HomeDirectory	Gets or sets the home directory for the pending import user.
ID	Gets the unique ID of the pending import user object.
Name	Gets or sets the UNIX user name for a pending import user.
PrimaryGroupID	Gets or sets the UNIX group identifier (GID) of the primary group for the pending import user profile.
Shell	Gets or sets the default login shell for the pending import user.
Source	Gets the text string that describes the source of the pending import data.
Status	Gets the status of the pending import user.
StatusDescription	Gets a text string that provides detailed information about the status of the pending import user.
TimeStamp	Gets the date and time that the pending user profiles were imported from the data source.

UID	Gets or sets the UNIX user identifier (UID) for the pending import user profile.

Commit

Commits any changes or updates to the pending import user object and saves them in Active Directory.

Syntax

```
void Commit()
```

Delete

Marks the pending user profile object for deletion from Active Directory.

Syntax

```
void Delete()
```

Discussion

This method does not delete the pending user profile. After you mark the object for deletion, you must use the Commit method to commit changes to the object to Active Directory. When the Commit method is executed, the pending user profile is deleted from Active Directory to complete the operation.

Exceptions

Delete throws an `UnauthorizedAccessException` if you have insufficient access rights to remove the UNIX profile in the zone. s

GetCandidate

Returns the directory object of a user pending import.

Syntax

```
DirectoryEntry GetCandidate()
```

Return value

The directory object of a user that is a candidate for import. Returns `null` if the candidate cannot be found.

Import

Imports the pending import user profile by associating the UNIX properties for the user with the specified Active Directory user account.

Syntax

```
void Import(IUser user)
```

Parameter

Specify the following parameter when using this method:

user	The user for which you want to retrieve profile information.

Discussion

This method links the pending import user to an Active Directory account and removes the user from the pending import list.

SetCandidate

Sets the directory object of a user pending import.

Syntax

```
void SetCandidate(DirectoryEntry entry)
```

Parameters

Specify the following parameter when using this method.

entry	The directory entry for the user that is a candidate for import.

Discussion

This method updates the CandidateDN property,

UpdateStatus

Checks the Active Directory forest for matching or conflicting information that will allow or prevent a pending import user being imported.

Syntax

```
void UpdateStatus()
```

Discussion

This method searches Active Directory for a user name that matches the pending import user name and updates the pending import user properties with the results of the search. For example, if no Active Directory match is found or a UNIX profile already exists for the matching Active Directory user, the method updates the pending user's properties with that information.

> **Note**: Checking the Active Directory forest for potential matching candidates or conflicts can be a time-intensive operation. Therefore, you should consider the size and distribution of the forest and limit the number of pending import users you are working with when using this method.

CandidateDN

Gets or sets the distinguished name (DN) of the import candidate.

Syntax

```
string CandidateDN {get; set;}
```

Property value

The matching Active Directory user object for pending user profile, if one is found. If there's no matching candidate in Active Directory, nothing is returned.

Discussion

This property returns the Active Directory user account that appears to match the pending user profile. If there's an existing Active Directory user that matches the pending user, the pending import user can be mapped to that account. If no matching candidate is found in Active Directory, this property returns a null value.

Gecos

Gets or sets the GECOS field of the UNIX profile for the pending import user.

Syntax

```
string Gecos {get; set;}
```

Property value

The text string value of the GECOS field in the UNIX profile for the pending import user.

HomeDirectory

Gets or sets the home directory field of the UNIX profile for the pending import user.

Syntax

```
string HomeDirectory {get; set;}
```

Property value

The text string value of the home directory field in the UNIX profile for the pending import user.

ID

Gets the unique ID of the pending import user object.

Syntax

```
string ID {get;}
```

Property value

The unique ID for the pending import group object.

Name

Gets or sets the UNIX user name for a pending import user.

Syntax

```
string Name {get; set;}
```

Property value

The UNIX user name of a pending import user.

PrimaryGroupID

Gets or sets the UNIX group identifier (GID) of the primary group for the pending import user profile.

Syntax

```
int PrimaryGroupID {get; set;}
```

Property value

The UNIX group identifier (GID) of the primary group for the pending import user profile.

Discussion

There are two versions of this property: one designed for COM-based programs that supports a 32-bit signed number one designed for .NET-based programs that allows a 64-bit signed number. Therefore, the data type for the property can be an integer (`int`) or a long integer (`long`) depending on the programming language you use.

Shell

Gets or sets the default login shell for the pending import user.

Syntax

```
string Shell {get; set;}
```

Property value

The text string value of the default login shell in the UNIX profile for the pending import user.

Source

Gets the text string that describes the source of the pending import data.

Syntax

```
string Source {get;}
```

Property value

A text string that describes the source of the pending import data.

Discussion

If the pending data was imported from a file, the property returns the source as File: followed be the path to the file name imported. If the source of the pending import data was a NIS server, the property returns the NIS server name and domain. For example, if the source of the data was a file, the property returns a string similar to this:

```
File: C:\\Migration\\magnolia_passwd
```

Status

Gets the status of the pending import user.

Syntax

```
StatusType Status {get;}
```

Property value

The status message for the pending import user.

Discussion

The status is determined by checking Active Directory for existing users that match or conflict with the pending import user. The property returns a number that determines the icon displayed for the user in the console. The icons indicate whether a pending user is:

Ready to import	Info	
Has potential issues that should be resolved	Warning	
Cannot be imported	Error	

StatusDescription

Gets a text string that provides detailed information about the status of the pending import user.

Syntax

```
string StatusDescription {get;}
```

Property value

The detailed status message for the pending import user.

Discussion

The status is determined by checking Active Directory for existing users that match or conflict with the pending import user. The results are displayed in **Access Manager** and in the **Status** tab of a pending user's **Properties** dialog box. The status description can also include details about the user's primary group. For example, if checking the Active Directory forest revealed a user name or UID conflict with an existing user or another pending import user, the StatusDescription property might include information similar to this:

```
No group with the corresponding GID found in Active Directory.
There is another pending imported user using the same UID.
There is another pending imported user using the same user name.
```

TimeStamp

Gets the date and time that the pending user profiles were imported from the data source.

Syntax

```
DateTime TimeStamp {get;}
```

Property value

The date and time that the pending user data was imported.

Example

The following code sample illustrates using this property in a script:

```
'Specify the zone you want to work with
Set objZone = cims.GetZone("w2k3.net/Acme/Zones/default")
‘Display the time users where imported
Set objPendUsers = objZone.GetImportPendingUsers
If not objPendUsers is nothing then
wScript.Echo "Imported from source: ", objPendUsers.TimeStamp
End if
…
```

UID

Gets or sets the UNIX user identifier (UID) for the pending import user profile.

Syntax

```
int UID {get; set;}
```

Property value

The UNIX user identifier (UID) for the pending import user profile.

Discussion

There are two versions of this property: one designed for COM-based programs that supports a 32-bit signed number one designed for .NET-based programs that allows a 64-bit signed number. Therefore, the data type for the property can be an integer (int) or a long integer (long) depending on the programming language you use.

UserInfos

The UserInfos class contains methods and properties used to manage a collection of pending import user profiles. This class is defined in the Centrify.DirectControl.API.Import namespace.

Syntax

```
public interface IUserInfos : IDisposable
```

Methods

The UserInfos class provides the following methods:

Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. Inherited from IDisposable.
Find	Returns the pending import user with the specified identifier from the collection of pending user profiles.
GetEnumerator	Returns an enumeration of UserInfo objects.

Properties

The UserInfos class provides the following properties:

Count	Gets the total number of pending import user profiles defined in the collection represented by the UserInfos object.
IsEmpty	Indicates whether the collection of pending import users is empty.

Find

Returns the pending import user with the specified identifier from the collection of pending user profiles.

Syntax

```
IUserInfo Find(string id)
```

Parameter

Specify the following parameter when using this method:

id	The unique identifier of the pending user profile for which you want to retrieve information.

Return value

The UserInfo object for the specified pending import user.

GetEnumerator

Returns an enumeration of IUserInfo objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of UserInfo objects.

Count

Gets the total number of pending import user profiles defined in the collection represented by the UserInfos object.

Syntax

```
int Count {get;}
```

Property value

The number of pending import user profiles in the set.

Discussion

This property enumerates all of the profiles in the collection before it returns the Count value. If you only need to determine whether any pending import groups, you should use the [IsEmpty](property for a faster response time.

IsEmpty

Determines whether the collection of pending import user profiles is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns `true` if there are no pending import user profiles in the `UserInfos` object, or `false` if there is at least one pending import user profile in the object.

Discussion

If there are no pending import user profiles in the `UserInfos` object, this property returns `true`. If these is at least one pending import user profile, the property returns `false`. Unlike the `Count` property, the `IsEmpty` property does not enumerate all of the pending import profiles in the collection before it returns a value. If you only need to determine whether any profiles are defined, you should call this property for a faster response.

UserUnixProfile

The UserUnixProfile class is an information class for managing user information in a specific zone.

Syntax

```
public interface IUserUnixProfile
```

Discussion

A user's zone-specific UNIX profile includes the numeric UID value, numeric GID value, default login shell, and default home directory. The GID can be associated with a standard Active Directory group, a standalone UNIX-only group profile not associated with any Active Directory group, and a local user profile.

Methods

The UserUnixProfile class provides the following methods:

Commit	Commits changes to the user profile to Active Directory.
Delete	Marks the user profile for deletion from Active Directory.
GetDirectoryEntry	Returns the directory entry for the UNIX user profile from Active Directory.
GetPrimaryGroup	Returns the UNIX profile of the primary group of the user.
Refresh	Reloads cached object data from Active Directory.
Validate	Checks whether the user profile contains valid data and can be committed to Active Directory.

Properties

The UserUnixProfile class provides the following properties:

ADsPath	Gets the LDAP path to the UNIX data object.
Cims	Gets the Cims object for the user.
HomeDirectory	Gets or sets the home directory for the user.
ID	Gets the unique identifier for the UserUnixProfile data object.
IsForeign	Indicates whether the Active Directory user associated with a UNIX profile is defined in a different forest than the zone (not applicable to local user profiles).
IsOrphan	Indicates whether this UNIX user is not associated with a corresponding Active Directory user (not applicable to local user profiles).
IsReadable	Indicates whether the Active Directory object is readable.
IsSFU	Indicates whether this UNIX user is an SFU zone profile (not applicable to local user profiles).
IsWritable	Determines whether the Active Directory object is writable.

Name	Gets or sets the UNIX login name of the user.
PrimaryGroup	Gets or sets the GID of the user's primary group.
ProfileState	Gets or sets the profile state of the local user profile (local user profiles only).
Shell	Gets or sets the default shell for the user.
Type	Gets the UserUnixProfile type for the user.
UnixEnabled	Determines whether the user's UNIX profile is enabled for access to the zone.
User	Gets the user object associated with this user UNIX profile (not applicable to local user profiles).
UserId	Gets or sets the UNIX user identifier (UID).
Zone	Gets the zone object associated with the user.

Commit

Commits changes to the user profile object and saves the changes in Active Directory.

Syntax

```
void Commit()
```

Discussion

If you are creating a new UNIX profile or updating a user's primary group or other attributes, you must use this method to complete the operation. If you have marked an object for deletion, this method deletes the object from Active Directory.

Exceptions

`Commit` may throw one of the following exceptions:

- `UnauthorizedAccessException` if your permissions are not sufficient to commit the Active Directory data object.
- `COMException` if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using `Commit` in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Set the UNIX profile for the user
set profile = objUser.SetUnixProfile(objZone, 10001, "pat", "/bin/bash",
"/home/pat", 10001)
'Validate the user's UNIX profile
profile.Validate
profile.Commit
...
```

Delete

Marks the user profile object for deletion from Active Directory.

Syntax

```
void Delete()
```

Discussion

This method does not delete the user profile. After you mark an object for deletion, you must use the Commit method to commit changes to the user object to Active Directory. When the Commit method is executed, the UNIX profile is deleted from Active Directory to complete the operation.

Example

The following code sample illustrates using Delete in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Mark the user profile for deletion
profile.Delete
...
```

GetDirectoryEntry

Returns an instance of the directory entry for the user's UNIX profile from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry()
```

Return value

A directory entry for the service connection point that represents the user's UNIX profile in the zone.

Discussion

This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

GetPrimaryGroup

Returns the UNIX profile of the primary group of the user.

Syntax

IGroupUnixProfile GetPrimaryGroup()

Return value

The UNIX profile of the user's primary group.

Example

The following code sample illustrates using GetPrimaryGroup in a script:

```
…
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Display the LDAP path for the user
wScript.Echo “Primary GID: “ & profile.GetPrimaryGroup().GID
…
```

Refresh

Reloads UNIX profile data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the UNIX profile information in the cached object to ensure it is synchronized with the latest information in Active Directory.

Example

The following code sample illustrates using Refresh in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfiles
'Reload the user's UNIX profile
profile.Refresh
...
```

Validate

Checks whether the user profile contains valid data and can be committed to Active Directory.

Syntax

```
void Validate()
```

Discussion

This method checks for errors in the UNIX profile fields and verifies that the profile includes a valid primary group, user name, UID, home directory, shell, and zone type. The method also verifies that the entry is not a duplicate of any existing profile in the zone. The method does not perform any of these checks, however, if the user profile is marked for deletion or if the profile has not been modified.

Exceptions

Validate throws an ApplicationException if the UNIX profile is missing data or contains invalid data.

Example

The following code sample illustrates using Validate in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Set the UNIX profile for the user
set profile = objUser.SetUnixProfile(objZone, 10001, "pat", "/bin/bash",
"/home/pat", 10001)
'Validate the user's UNIX profile
profile.Validate
...
```

ADsPath

Gets the LDAP path to the UNIX profile data object.

Syntax

```
string ADsPath {get;}
```

Property value

The LDAP path to the UNIX profile data object.

Example

The following code sample illustrates using ADsPath in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Display the LDAP for the user's UNIX profile
wScript.Echo "LDAP Path: " & profile.ADsPath
...
```

Cims

Gets the Cims object for the user.

Syntax

```
Cims Cims {get;}
```

Property value

The Cims object.

Discussion

This property serves as a shortcut for retrieving data.

HomeDirectory

Gets or sets the home directory for the user.

Syntax

```
string HomeDirectory {get; set;}
```

Property value

A string that defines the path to the default home directory for the user from the user's UNIX profile.

Example

The following code sample illustrates using HomeDirectory in a script:

```
…
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Change the default home directory in the user's UNIX profile
set profile.HomeDirectory = "/home/all_users/pathu"
…
```

ID

Gets the unique identifier for the UserUnixProfile data object.

Syntax

```
string ID {get;}
```

Property value

The unique identifier for the UserUnixProfile data object.

Example

The following code sample illustrates using ID in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Display the unique identifier for the user's UNIX profile
wScript.Echo "Unique ID: " & profile.ID
...
```

IsForeign

Indicates whether the corresponding Active Directory user for a UNIX profile is in a different Active Directory forest than the forest associated with the user's UNIX profile in the zone.

Syntax

```
bool IsForeign {get;}
```

Property value

Returns true if the UNIX profile is associated with an Active Directory user in a different forest.

Discussion

If the Active Directory user is in a different forest than the one associated with a top-level Cims object, the property returns true.

Example

The following code sample illustrates using IsForeign in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Check the forest for Users in the zone
For each profile in objZone.GetUserUnixProfiles
if profile.IsForeign then
wScript.Echo “Foreign user: ” & profile.Name
end if
next
...
```

IsOrphan

Indicates whether this UNIX user is not associated with a corresponding Active Directory user.

Syntax

```
bool IsOrphan {get;}
```

Property value

Returns `true` if the corresponding Active Directory user for a UNIX profile is not found, or `false` if the corresponding Active Directory user for the UNIX profile exists.

Discussion

This property can be used to determine whether the Active Directory user associated with a UNIX profile has been deleted from Active Directory.

Example

The following code sample illustrates using `IsOrphan` in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Check for Orphan Users in the zone
For each profile in objZone.GetUserUnixProfiles
if profile.IsOrphan then
wScript.Echo “Orphan user: ” & profile.Name
end if
next
...
```

IsReadable

Indicates whether the user profile object in Active Directory is readable for the current user credentials.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the UserUnixProfile object is readable, or false if the object is not readable.

Discussion

This property returns a value of true if the user accessing the user profile object in Active Directory has sufficient permissions to read its properties.

Example

The following code sample illustrates using IsReadable in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Check whether the user's UNIX profile is readable
if not profile.IsReadable then
wScript.Echo "No permission to read the UNIX profile!"
else
wScript.Echo "UNIX login name: " & profile.Name
end if
...
```

IsSFU

Indicates whether the UNIX user profile is in a Services for UNIX (SFU) zone.

Syntax

```bool IsSFU {get;}```

Property value

Returns true if the user profile is a Services for UNIX (SFU) profile.
```

IsWritable

Indicates whether the user profile object is writable for the current user's credentials.

Syntax

```
bool IsWritable {get;}
```

Property value

Returns `true` if the `UnixUserProfile` object is writable, or `false` if the object is not writable.

Discussion

This property returns a value of `true` if the user accessing the user profile object in Active Directory has sufficient permissions to change the user profile object's properties.

Example

The following code sample illustrates using `IsWritable` in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Check whether the user's UNIX profile is writable
if not profile.IsWritable then
wScript.Echo "No permission to change the UNIX profile!"
end if
...
```

Name

Gets or sets the UNIX login name of the user in the user's UNIX profile.

Syntax

```
string Name {get; set;}
```

Property value

The UNIX login name from the user's UNIX profile.

Example

The following code sample illustrates using Name in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Set the UNIX profile for the user
set profile = objUser.SetUnixProfile(objZone, 10001, "pat", "/bin/bash",
"/home/pat", 10001)
'Display the user's UNIX login name
profile.Name
...
```

PrimaryGroup

Gets or sets the group identifier (GID) of the primary group for the user.

Syntax

```
long PrimaryGroup {get; set;}
```

Property value

The value used as the GID for the user's primary group.

Discussion

This method is used internally by .NET modules.

Example

The following code sample illustrates using PrimaryGroup in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Set the GID for the user's primary group
Set objUser.PrimaryGroup = 490007
...
```

ProfileState

Gets the profile state of an existing local user or sets the profile of the specified local user.

Syntax

```
UserProfileState ProfileState{get; set;}
```

Property value

The profile state of the specified local user.

Exceptions

`ProfileState` throws an `InvalidOperationException` if the user you specify is not a local user.

Shell

Gets or sets the default shell for the user.

Syntax

```
string Shell {get; set;}
```

Property value

A string that defines the path to the default shell for the user from the user's UNIX profile.

Example

The following code sample illustrates using Shell in a script:

```
…
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Change the default shell in the user's UNIX profile
set profile.Shell = "bin/sh"
…
```

Type

Gets the user UNIX profile type for the user.

Syntax

```
UserUnixProfileType Type {get;}
```

Property value

A numeric value that indicates whether the UNIX profile is a standard Delinea profile or a Service for UNIX (SFU) profile.

Possible values:

```
public enum UserUnixProfileType
{
    // Centrify user
    Centrify = 0,
    // Microsft Service for Unix type
    Sfu = 1,
    // MIT Kerberos-realm trusted user
    MIT = 2
}
```

Discussion

There is no AD User object corresponding to an MIT user type of profile.

Example

The following code sample illustrates using Type in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Check the profile type in the user's UNIX profile
if profile.Type = 0
    wScript.Echo "Standard UNIX profile"
else
    wScript.Echo "SFU user UNIX profile"
end if
...
```

UnixEnabled

Determines whether the user's UNIX profile is enabled for access to the zone. This attribute is only applicable in classic zones and for backwards compatibility.

Syntax

```
bool UnixEnabled {get; set;}
```

Property value

Returns true if the user's UNIX profile is enabled for access in a classic zone, or false if the UNIX profile is not enabled for access to the zone.

Discussion

The UnixEnabled attribute determines whether a specific profile is enabled or disabled for access to the zone. If this property is set to true, the user's UNIX profile can be used to access the current zone. If this property is set false, the user cannot log on to computers in the zone using the disabled UNIX profile.

Exceptions

UnixEnabled throws an InvalidOperationException if the user is assigned to a hierarchical zone.

Example

The following code sample illustrates using UnixEnabledv in a script:

```
...
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the user object
set objUser = cims.GetUserByPath("LDAP://CN=pat.hu,CN=Users,DC=ajax,DC=org")
'Disable the user's UNIX profile in this zone
objUserUnixProfile.UnixEnabled = False
...
```

User

Gets the Active Directory user object associated with this user UNIX profile.

Syntax

IUser User {get;}

Property value

The user object.

UserId

Gets or sets the UID of the user.

Syntax

```
long UserId {get; set;}
```

Property value

The UID of the user.

Zone

Gets the zone object associated with the user.

Syntax

IZone Zone {get;}

Property value

The zone object associated with the user.

Example

The following code sample illustrates using Zone in a script:

```
…
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
'Get the Active Directory user object
set objUser = cims.GetUser("ajax.org/Users/pat.hu")
'Get the UNIX profile for the user
profile = objUser.UnixProfileByUid(10001)
'Display the zone associated with the user's UNIX profile
wScript.Echo "Zone: " & profile.Zone
…
```

UserUnixProfiles

The UserUnixProfiles class is used to manage a collection of user profiles.

Syntax

```
public interface IUserUnixProfiles
```

Discussion

The collection of user profiles contained in the object depend on how the object was obtained:

- When you call User.UnixProfiles, the UserUnixProfiles object returned enumerates all of the profiles defined for a specific Active Directory user across all zones in the current domain.
- When you call Zone.GetUserUnixProfiles, the UserUnixProfiles object returned enumerates all of the profiles defined for a specific Active Directory user in a specific zone.

Methods

The UserUnixProfiles class provides the following methods:

GetEnumerator	Returns an enumeration of user profiles.
Refresh	Reloads the collection of UNIX profiles from Active Directory.

Properties

The UserUnixProfiles class provides the following properties:

Count	Gets the number of UNIX profiles defined in the collection of UserUnixProfiles.
IsEmpty	Indicates whether the UserUnixProfiles object contains any profiles.

GetEnumerator

Returns an enumeration of user profiles.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

The set of user profile objects.

Refresh

Reloads the collection of UNIX profiles from Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the collections of UNIX profiles in the cached object to ensure the object is synchronized with the latest information in Active Directory.

Count

Determines the total number of UNIX profiles defined in the UserUnixProfiles collection.

Syntax

```
int Count {get;}
```

Property value

The number of UNIX profiles in the set.

Example

The following code sample illustrates using Count in a script:

```
…
'Get the zone object
Set objZone = cims.GetZone("ajax.org/UNIX/Zones/pilot")
Set objUserUnixProfiles = objUser.UnixProfiles
wScript.Echo "Profile count: " & objUserUnixProfiles.Count
…
```

IsEmpty

Determines whether the collection of UNIX user profiles is empty.

Syntax

```
bool IsEmpty {get;}
```

Property value

Returns `true` if there are no user profiles in the `UserUnixProfiles` object.

Discussion

Unlike the `Count` property, the `IsEmpty` property does not query all of the profiles in the collection before it returns a value. If you only need to determine whether any profiles are defined, you should call this property for a faster response.

WindowsApplication

This class represents a Windows application right.

Syntax

public interface IWindowsApplication:IRight

Methods

The WindowsApplication class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Creates the match criteria used to identify the Windows application to which you want to grant a right.
Unexpected Link Text	Removes the right. (Inherited from Unexpected Link Text.)

Properties

The WindowsApplication class provides the following properties:

Unexpected Link Text	Gets or sets the properties that are used to identify a specific Windows application.
Unexpected Link Text	Gets or sets the description of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the priority of this right.
Unexpected Link Text	Gets or sets whether the user's password is required when this right is used.
Unexpected Link Text	Gets or sets the SID for the run-as user or an SID for the user assigned the right (VBScript).
Unexpected Link Text	Gets or sets the SID for the run-as user or a list of SIDs for the users assigned the right (.NET).
Unexpected Link Text	Gets or sets the run-as type for the right.
Unexpected Link Text	Gets the zone this right belongs to. (Inherited from Unexpected Link Text.)

Discussion

A Windows application right enables a user to run a Windows application with the privileges of another user or as a member of an Active Directory or built-in group. For example, you can use a Windows application right to give a standard Windows user elevated privileges to run an database management application as a database administrator.

CreateApplicationCriteria

Creates the match criteria to identify the Windows application and related requirements to which you want to grant a right.

Syntax

```
void CreateApplicationRight()
```

Discussion

This method initiates the collection of the set of match criteria, defined using the properties of the WindowsApplicationCriteria class, to identify a specific Windows application to which you want to grant a right.

Example

See Unexpected Link Text for a code sample that illustrates using CreateApplicationRight in a script.

ApplicationCriteriaList

Gets or sets the list of properties that are used to identify a specific Windows application.

Syntax

In .NET:

```
IEnumerable\<IWindowsApplicationCriteria\> ApplicationCriteriaList {get; set;}
```

In VBScript:

```
object[] ApplicationCriteriaList {get; set;}
```

Property value

The complete match criteria defined to identify a specific Windows application.

Example

The following code sample illustrates using ApplicationCriteriaList in a script:

```
// Create a new Windows application right with some basic properties.
$objWindowsApplication = $objZone.CreateWindowsApplication();
$objWindowsApplication.Name = $strWindowsApplication;
$objWindowsApplication.RunAsType = $runAsType;
$objWindowsApplication.RunAsString = $strDnList;
$objWindowsApplication.RequirePassword = $requirePassword;
$objWindowsApplication.Description = "optional description";
$objWindowsApplication.Priority = 0;
// Specify the criteria used to identify the Windows application.
$listType = ("System.Collections.Generic.List\`1" -as "Type");
$listType = $listType.MakeGenericType( @(
("Centrify.DirectControl.API.IWindowsApplicationCriteria" -as "Type")));
$criteriaList = [Activator]::CreateInstance($listType);
$objApplicationCriteria = $objWindowsApplication.CreateApplicationCriteria();

$objApplicationCriteria.FileType =
[Centrify.DirectControl.API.WindowsFileType]::EXE;
$objApplicationCriteria.FileName = "calc.exe";
$objApplicationCriteria.Path = "SYSTEMPATH";
$objApplicationCriteria.FileDescription = "Windows Calculator";
$objApplicationCriteria.FileDescriptionMatchOption =
[Centrify.DirectControl.API.StringMatchOption]::ExactMatch;
$objApplicationCriteria.FileVersion = "6.1";
$objApplicationCriteria.FileVersionMatchOption =
[Centrify.DirectControl.API.VersionMatchOption]::LaterThanOrEqualTo;
$objApplicationCriteria.Description = "Match criteria for Windows Calc";
$objWindowsApplication.ApplicationCriteriaList = $criteriaList;
$objWindowsApplication.Commit();
Write-Host("WindowsApplication {0} has been added to zone {1} successfully." -f
$strWindowsApplication, $strZone);
exit 0;
}
```

Priority

Gets or sets the priority of this right.

Syntax

```
int Priority {get; set;}
```

Property value

The priority of the right. Default is 0.

Discussion

This number is used when handling multiple matches for rights specified by wild cards. If rights specified by this property object match rights specified by another property object, the object with the higher priority prevails. The higher the value of the Priority property, the higher the priority.

Example

See Unexpected Link Text for a code sample that illustrates using Priority in a script.

RequirePassword

Gets or sets whether the logged-in user's password is required when this right is used.

Syntax

```
bool RequirePassword {get; set;}
```

Property value

Set to true if the right requires the logged-in user's password.

Example

See Unexpected Link Text for a code sample that illustrates using RequirePassword in a script.

RunAs

Gets or sets the run-as property for this right.

Syntax

```
string RunAs {get; set;}
```

Property value

The run-as property for a single user.

Discussion

If the Unexpected Link Text property is set to Self, the remote application is run under the logged-in user account, but with the additional privileges of the user whose SID is listed in the RunAs property. For example, if the WindowsApplication right is set to run as Self and RunAs contains the SID of the Network Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Network Admins group.

If the RunAsType property is set to User, the remote application is run under the user whose SID is listed in the RunAs property. For example, if the WindowsApplication right is set to run as User and RunAs contains the SID of the user NetAdmin, then this application runs with the permissions of the NetAdmin user.

If the RunAs property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

> **Note**: This property is for use in VBScript programs. Use the RunAsList property for .NET.

RunAsList

Gets or sets the run-as list for this right.

Syntax

```
IList\<SecurityIdentifier\> RunAsList {get; set;}
```

Property value

The run-as list for the right.

Discussion

If the Unexpected Link Text property is set to Self, the application is run under the logged-in user account, but with the additional privileges of the groups whose SIDs are listed in the RunAsList property. For example, if the WindowsApplication right is set to run as Self and RunAsList contains the SID of the Local Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Local Admins group.

If the RunAsType property is set to User, the application is run under the user whose SID is listed in the RunAsList property. In this case, the RunAsList property contains only a single SID. For example, if the WindowsApplication right is set to run as User and RunAsList contains the SID of the user Admin, then this application runs with the permissions of the Admin user.

If the RunAsList property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

> **Note**: This property can only be used in .NET programs. Use the RunAs property for VBScript.

RunAsType

Gets or sets the run-as type for this right.

Syntax

```
WindowsRunAsType RunAsType {get; set;}
```

Property value

The run-as type of the right.

Possible values:

```
public enum WindowsRunAsType
{
    // Run as self
    Self,
    // Run as another user
    User
}
```

Discussion

If the Unexpected Link Text property is set to Self, the application runs as the logged-in user with the additional privileges of the groups whose SIDs are listed in the RunAsList property. For example, if the WindowsApplication right is set to run as Self and RunAsList contains the SID of the Local Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Local Admins group.

If the RunAsType property is set to User, the application is run as the user whose SID is listed in the [RunAsList(../networkaccess/runaslist.md) property. For example, if the WindowsApplication right is set to run as User and RunAsList contains the SID of the user Admin, then this application runs as Admin with the permissions of that user.

Example

See ApplicationCriteriaList for a code sample that illustrates using RunAsType in a script.

WindowsApplicationCriteria

This class represents the match criteria for identifying a Windows application right.

Syntax

```
public interface IWindowsApplicationCriteria
```

Methods

The WindowsApplicationCriteria class provides the following method.

Unexpected Link Text	Determines whether the match criteria is valid.

Properties

The WindowsApplicationCriteria class provides the following match criteria properties that correspond to the fields displayed on the Match Criteria tab in Access Manager:

Unexpected Link Text	Gets or sets the arguments allowed with this Windows application right.
Unexpected Link Text	Gets or sets the company name to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the string specified for the CompanyName property must be an exact match or a partial match.
Unexpected Link Text	Gets or sets the description for the match criteria defined for a specific application right.
Unexpected Link Text	Gets or sets the file description to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the string specified for the FileDescription property must be an exact match or a partial match.
Unexpected Link Text	Gets or sets the encrypted file hash to match to identify the Windows application associated with this right. The file hash is generated using the SHA-1 encryption algorithm, which is FIPS-compliant.
Unexpected Link Text	Gets or sets the file name to match to identify the Windows application associated with this right.
Unexpected Link Text	Gets or sets the type of executable file to match to identify the Windows application associated with this right.
Unexpected Link Text	Gets or sets the file version to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the version must be equal to, earlier than or equal to, or later than or equal to the version specified for the FileVersion property.

Unexpected Link Text	Specifies whether arguments for the Argument property are case-sensitive.
Unexpected Link Text	Specifies whether the string specified for the Argument property must be an exact match or a partial match.
Unexpected Link Text	Gets or sets the local user name or group name to match to allow the use of this Windows application right.
Unexpected Link Text	Specifies the distinguished name of the Active Directory user or group who is the file owner to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies the security identifier of the Active Directory user or group who is the file owner to match to identify the Windows application associated with this right.
Unexpected Link Text	Gets or sets the type of owner to match to allow the use of this Windows application right.
Unexpected Link Text	Gets or sets the path to the executable to match to identify the Windows application associated with this right.
Unexpected Link Text	Gets or sets the product name to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the string specified for the ProductName property must be an exact match or a partial match.
Unexpected Link Text	Gets or sets the product version to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the version must be equal to, earlier than or equal to, or later than or equal to the version specified for the ProductVersion property.
Unexpected Link Text	Gets or sets the publisher name to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the publisher must be an exact match, partial match, start with, or end with the string specified for the Publisher property.
Unexpected Link Text	Specifies whether the Windows application associated with this right requires an administrative user account.
Unexpected Link Text	Gets or sets the volume serial number to match to identify the Windows application associated with this right.
Unexpected Link Text	Specifies whether the volume serial number must be an exact match, partial match, start with, or end with the string specified for the SerialNumber property.

Discussion

A Windows application right enables a user to run a Windows application with the privileges of another user or as a member of an Active Directory or built-in group. For example, you can use a Windows application right to give a standard Windows user elevated privileges to run an database management application as a database administrator. You can define the criteria to use to identify the Windows application associated with an application right using the WindowsApplicationCriteria properties.

Validate

Determines whether the match criteria is valid.

Syntax

```
void Vallidate()
```

Discussion

This property is only applicable in .NET-compatible programs.

Exception

`Validate` throws an `ApplicationException` if any property specified for the match criteria is not valid.

Argument

Gets or sets the arguments allowed with this Windows application right.

If you specify a file type of .msc, you must also specify the Arguments property. The Arguments property is optional for all other file types. If this property is set to an empty string, no arguments are allowed. Do not specify the Arguments property if you are granting a right to access the application without argument restrictions. If the property is not defined, the application can be executed with any argument or combination of arguments.

Syntax

```
string Argument {get; set;}
```

Property value

A list of arguments that can be used with the application when this application is executed.

CompanyName

Gets or sets the company name to match to identify the Windows application associated with this right.

Syntax

```
string CompanyName {get; set;}
```

Property value

The company name associated with the application.

Example

The following code sample illustrates using CompanyName in a script:

```
...
\$listType = ("System.Collections.Generic.List\``1" -as "Type");
\$listType = \$listType.MakeGenericType( @(
("Centrify.DirectControl.API.IWindowsApplicationCriteria" -as "Type")));
\$criteriaList = [Activator]::CreateInstance(\$listType);
\$objApplicationCriteria = \$objWindowsApplication.CreateApplicationCriteria();

\$objApplicationCriteria.FileType =
[Centrify.DirectControl.API.WindowsFileType]::EXE;
\$objApplicationCriteria.FileName = "filename.exe";
\$objApplicationCriteria.Path = "SYSTEMPATH";
\$objApplicationCriteria.Argument = "Optional arguments 1";
\$objApplicationCriteria.IsArgumentCaseSensitive = \$true;
\$objApplicationCriteria.IsArgumentExactMatch = \$true;
\$objApplicationCriteria.CompanyName = "Cendura Software";
\$objApplicationCriteria.CompanyNameMatchOption =
[Centrify.DirectControl.API.StringMatchOption]::ExactMatch;
...
```

CompanyNameMatchOption

Specifies whether the string specified for the CompanyName property must be an exact match or can be a partial match to identify an application associated with this right.

Syntax

```
StringMatchOption CompanyNameMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the company name field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
    ExactMatch,
    // Partial match
    Contains
}
```

Example

See Unexpected Link Text for code sample that illustrates using CompanyNameMatchOption in a script.

Description

Gets or sets the description for the set of match criteria defined for a specific application right.

For example, if you are defining match criteria that will grant an application right for multiple versions of SQL Server Management Studio running on different versions of the Windows operating system, you might specify a description such as “SQL Server Management Studio (2005-2012)” to indicate the scope of the right.

Syntax

```
string Description {get; set;}
```

Property value

The descriptive text for a set of match criteria.

Example

The following code sample illustrates using Description in a script:

```
\$objWindowsApplication =
\$objZone.GetWindowsApplication(\$strWindowsApplication);
{
\$objWindowsApplication = \$objZone.CreateWindowsApplication();
\$objWindowsApplication.Name = \$strWindowsApplication;
\$objWindowsApplication.RunAsType = \$runAsType;
\$objWindowsApplication.RunAsString = \$strDnList;
\$objWindowsApplication.RequirePassword = \$requirePassword;
\$objWindowsApplication.Description = "SQL Server Match criteria";
\$objWindowsApplication.Priority = 0;
…
```

FileDescriptionMatchOption

Specifies whether the string specified for the FileDescription property must be an exact match or a partial match to identify an application associated with this right.

Syntax

```
StringMatchOption FileDescriptionMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the file description field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
    ExactMatch,
    // Partial match
    Contains
}
```

FileDescriptionMatchOption

Specifies whether the string specified for the FileDescription property must be an exact match or a partial match to identify an application associated with this right.

Syntax

```
StringMatchOption FileDescriptionMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the file description field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
    ExactMatch,
    // Partial match
    Contains
}
```

FileHash

Gets or sets the encrypted file hash to match to identify the Windows application associated with this right. The file hash is generated using the SHA-1 encryption algorithm, which is FIPS-compliant.

Syntax

```
string FileHash {get; set;}
```

Property value

The file hash to match to identify the application.

FileName

Gets or sets the file name to match to identify the Windows application associated with this right.

Syntax

```
string FileName {get; set;}
```

Property value

The file name to match to identify the application.

FileType

Gets or sets the type of executable file to match to identify the Windows application associated with this right. You must specify a file type to define a valid application right.

Syntax

```
WindowsFileType FileType {get; set;}
```

Property value

The executable file type to match.

Possible values:

```
public enum WindowsFileType
{
    // Batch file
       BAT,
    // Command script
       CMD,
    // Command file
       COM,
    // Control Panel Extension
       CPL,
    // Executable file
       EXE
    // Microsoft common console document
       MSC
    // Windows installer package
       MSI
    // Windows installer patch
       MSP
    // Windows PowerShell cmdlet
       PS1
    // VBScript script
       VBS
    // Windows script file
       WSF
}
```

Example

The following code sample illustrates using FileType in a script:

```
$objWindowsApplication =
$objZone.GetWindowsApplication($strWindowsApplication);
{
...
$listType = $listType.MakeGenericType( @
("Centrify.DirectControl.API.IWindowsApplicationCriteria" -as "Type")));
$criteriaList = [Activator]::CreateInstance($listType);
$objApplicationCriteria = $objWindowsApplication.CreateApplicationCriteria();

$objApplicationCriteria.FileType =
[Centrify.DirectControl.API.WindowsFileType]::EXE;
$objApplicationCriteria.FileName = "filename.exe";
...
}
```

FileVersion

Gets or sets the file version to match to identify the Windows application associated with this right.

Syntax

```
string FileVersion {get; set;}
```

Property value

The file version to match to identify the application.

FileVersionMatchOption

Specifies whether the version must be equal to, earlier than or equal to, or later than or equal to the version specified for the FileVersion property.

Syntax

```
VersionMatchOption FileVersionMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the file version field.

Possible values:

```
public enum VersionMatchOption
{
    // Match the version specified
       Equal,
    // Match earlier than or equal to the specified version
       EarlierThanOrEqualTo,
    // Match later than or equal to the specified version
       LaterThanOrEqualTo
}
```

IsArgumentCaseSensitive

Specifies whether the arguments for the Argument property are case-sensitive.

Syntax

```
bool IsArgumentCaseSensitive {get; set;}
```

Property value

Set to true if arguments are case-sensitive. The default is false.

IsArgumentExactMatch

Specifies whether the arguments specified for the Argument property must be an exact match.

Syntax

```
bool IsArgumentExactMatch {get; set;}
```

Property value

Set to true if the arguments must be an exact match. The default is false.

LocalOwner

Gets or sets the local owner to match to allow the use of this Windows application right.

Syntax

```
string LocalOwner {get; set;}
```

Property value

The local owner user name or group to match to execute the application.

OwnerDN

Specifies the distinguished name of the Active Directory user or group who is the file owner to match to identify the Windows application associated with this right.

This property is only applicable in VBScript programs.

Syntax

```
string OwnerDN {get; set;}
```

Property value

The distinguished name of the owner of the file to match.

OwnerSid

Specifies the security identifier of the Active Directory user or group who is the file owner to match to identify the Windows application associated with this right.

This property is only applicable in .NET-compatible programs.

Syntax

```
SecurityIdentifier OwnerSid {get; set;}
```

Property value

The security identifier of the owner of the file to match.

OwnerType

Gets or sets the type of owner to match to allow the use of this Windows application right.

Syntax

In .NET:

```
Nullable\<WindowsFileOwnerType\> OwnerType {get; set;}
```

In VBScript:

```
WindowsFileOwnerType OwnerType {get; set;}
```

Property value

The type of owner to match.

Possible values:

```
public enum WindowsFileOwnerType
{
    // Active Directory group or user SID
    Sid,
    // Local group account
    LocalGroup,
    // Local user account
    LocalUser
}
```

Path

Gets or sets the path to the executable to match to identify the Windows application associated with this right.

Syntax

```
string Path {get; set;}
```

Property value

Path to the executable for the application. You can specify multiple paths separated by semicolons (;). You can specify the standard system path using the SYSTEMPATH keyword, specify a full custom path, such as C:\\Windows\\system32\\inetsrv, or use one of the supported path variables.

The supported path variables are:

- %systemroot%
- %system32%
- %syswow64%
- %program files%
- %program files(x86)%

The space between “program” and “files” is required.

ProductName

Gets or sets the product name to match to identify the Windows application associated with this right.

Syntax

```
string ProductName {get; set;}
```

Property value

The product name to match to identify the application.

ProductNameMatchOption

Specifies whether the string specified for the ProductName property must be an exact match or a partial match.

Syntax

```
StringMatchOption ProductNameMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the product name field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
       ExactMatch,
    // Partial match
       Contains
}
```

ProductVersion

Gets or sets the product version to match to identify the Windows application associated with this right.

Syntax

```
string ProductVersion {get; set;}
```

Property value

The product version to match to identify the application.

ProductVersionMatchOption

Specifies whether the version must be equal to, earlier than or equal to, or later than or equal to the version specified for the ProductVersion property.

Syntax

```
VersionMatchOption ProductVersionMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the product version field.

Possible values:

```
public enum VersionMatchOption
{
    // Match the version specified
       Equal,
    // Match earlier than or equal to the specified version
       EarlierThanOrEqualTo,
    // Match later than or equal to the specified version
       LaterThanOrEqualTo
}
```

Publisher

Gets or sets the publisher information from a digital certificate to match to identify the Windows application associated with this right.

Syntax

```
string Publisher {get; set;}
```

Property value

The publisher information to match to identify the application.

PublisherMatchOption

Specifies whether the publisher information must be an exact match, partial match, start with, or end with the string specified for the Publisher property.

Syntax

```
StringMatchOption PublisherMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the publisher field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
       ExactMatch,
    // Partial match
       Contains
    // Starts with match
       StartsWith
    // Ends with match
       EndsWith
}
```

RequireAdministrator

Specifies whether the Windows application associated with this right requires an administrative user account.

Syntax

```
bool RequireAdministrator {get; set;}
```

Property value

Set to true if the application must be executed using an administrative user account. The default is false.

SerialNumber

Gets or sets the volume serial number to match to identify the Windows application associated with this right.

This property is used to match an application located on a CD or DVD media with the specified volume serial number. If the target application is not executed from CD or DVD media, this property does not apply and the application right will not be granted.

Syntax

```
string SerialNumber {get; set;}
```

Property value

The volume serial number information to match to identify the application.

SerialNumberMatchOption

Specifies whether the volume serial number must be an exact match, partial match, start with, or end with the string specified for the SerialNumber property.

Syntax

```
StringMatchOption SerialNumberMatchOption {get; set;}
```

Property value

The match criteria operator, which specifies the type of matching to perform for the volume serial number field.

Possible values:

```
public enum StringMatchOption
{
    // Exact match
       ExactMatch,
    // Partial match
       Contains
    // Starts with match
       StartsWith
    // Ends with match
       EndsWith
}
```

WindowsApplications

The WindowsApplications class manages a collection of Windows application rights.

Syntax

```
public interface IWindowsApplications
```

Methods

The WindowsApplications class provides the following method:

Unexpected Link Text	Gets the enumerator you can use to enumerate all windows application rights.

GetEnumerator

Returns an enumeration of WindowsApplication objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all of the WindowsApplication objects.

Delinea

WindowsDesktop

This class represents a Windows desktop right.

Syntax

public interface IWindowsDesktop:IRight

The WindowsDesktop class provides the following methods:

Unexpected Link Text	Commits changes in the right to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Removes the right. (Inherited from Unexpected Link Text.)

Properties

The WindowsDesktop class provides the following properties:

Unexpected Link Text	Gets or sets the description of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is readable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the right is writable. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the right. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the priority of this right.
Unexpected Link Text	Gets or sets whether the user's password is required when this right is used.
Unexpected Link Text	Gets or sets the SID for the run-as user or an SID for the users assigned the right (VBScript).
Unexpected Link Text	Gets or sets the SID for the run-as user or a list of SIDs for the users assigned the right (.NET).
Unexpected Link Text	Gets or sets the run-as type for the right.
Unexpected Link Text	Gets the zone this right belongs to. (Inherited from Unexpected Link Text.)

Discussion

A Windows desktop right provides a complete desktop that behaves as if the user had logged in as specific privileged user with the privileges of another user or as a member of an Active Directory or built-in group. For example, you can use a Windows desktop right to give a standard Windows user elevated privileges to run local applications as a member of the built-in Administrators group.

Priority

Gets or sets the priority of this right.

Syntax

```
int Priority {get; set;}
```

Property value

The priority of the right. Default is 0.

Discussion

This number is used when handling multiple matches for rights specified by wild cards. If rights specified by this property object match rights specified by another property object, the object with the higher priority prevails. The higher the value of the Priority property, the higher the priority.

RequirePassword

Gets or sets whether the logged-in user's password is required when this right is used.

Syntax

```
bool RequirePassword {get; set;}
```

Property value

Set to `true` if the right requires the logged-in user's password.

RunAs

Gets or sets the run-as property for this right.

Syntax

```
string RunAs {get; set;}
```

Property value

The run-as property for a single user.

Discussion

If the Unexpected Link Text property is set to Self, the remote application is run under the logged-in user account, but with the additional privileges of the groups whose SIDs are listed in the RunAs property as a semicolon (;) separated string. For example, if the WindowsDesktop right is set to run as Self and RunAs contains the SID of the Network Admins group, then this application runs with the permissions of the logged-in user plus the permissions of the Network Admins group.

If the RunAsType property is set to User, the remote application is run under the user whose SID is listed in the RunAs property. For example, if the WindowsDesktop right is set to run as User and RunAs contains the SID of the user NetAdmin, then this application runs with the permissions of the NetAdmin user.

If the RunAs property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

> **Note**: This property is for use in VBScript programs. Use the RunAsList property for .NET.

RunAsList

Gets or sets the run-as list for this right.

Syntax

```
IList\<SecurityIdentifier\> RunAsList {get; set;}
```

Property value

The run-as list for the right.

Discussion

If the Unexpected Link Text property is set to Self, the desktop runs as the logged-in user with the additional privileges of the groups whose SIDs are listed in the RunAsList property. For example, if the WindowsDesktop right is set to run as Self and RunAsList contains the SID of the Local Admins group, then this desktop runs with the permissions of the logged-in user plus the permissions of the Local Admins group.

If the RunAsType property is set to User, the desktop is run as the user whose SID is listed in the RunAsList property. In this case, the RunAsList property contains only a single SID. For example, if the WindowsDesktop right is set to run as User and RunAsList contains the SID of the user Admin, then this desktop runs as Admin with the permissions of that user.

If the RunAsList property is empty, this right is invalid and an exception is thrown when you call the Unexpected Link Text method.

> **Note**: This property can only be used in .NET programs. Use the RunAs property for VBScript.

RunAsType

Gets or sets the run-as type for this right.

Syntax

```
WindowsRunAsType RunAsType {get; set;}
```

Property value

The run-as type of the right.

Possible values:

```
public enum WindowsRunAsType
{
    // Run as self
    Self,
    // Run as another user
    User
}
```

Discussion

If the RunAsType property is set to Self, the desktop runs as the logged-in user with the additional privileges of the groups whose SIDs are listed in the Unexpected Link Text property. For example, if the WindowsDesktop right is set to run as Self and RunAsList contains the SID of the Local Admins group, then this desktop runs with the permissions of the logged-in user plus the permissions of the Local Admins group.

If the RunAsType property is set to User, the desktop is run as the user whose SID is listed in the RunAsList property. For example, if the WindowsDesktop right is set to run as User and RunAsList contains the SID of the user Admin, then this desktop runs as Admin with the permissions of that user.

WindowsDesktops

The WindowsDesktops class manages a collection of Windows desktop rights.

Syntax

```
public interface IWindowsDesktops
```

Methods

The WindowsDesktops class provides the following method:

Unexpected Link Text	Gets the enumerator you can use to enumerate all windows desktop rights.

GetEnumerator

Returns an enumeration of WindowsDesktop objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all the WindowsDesktop objects.

WindowsUser

The WindowsUser class manages the Windows user profile information of a user.

Syntax

```
public interface IWindowsUser
```

Methods

The WindowsUser class provides the following methods:

Unexpected Link Text	Adds a role assignment to the user profile.
Unexpected Link Text	Returns the directory entry for the user from Active Directory.
Unexpected Link Text	Returns an enumeration of the effective user role assignments.

Properties

The WindowsUser class provides the following property:

Unexpected Link Text	Gets the Windows login name of the user.

Discussion

A user's Windows profile consists of the information used by Windows to identify the user. See the HierarchicalUser class for a user's UNIX profile.

The rights you assign to users and group in a particular role apply to Active Directory users and groups. They can also apply to locally-defined users and groups if you configure the role definition to allow local accounts to be assigned to the role. All Windows users, including local users, must be assigned at least one role that allows them log on locally, remotely, or both.

AddUserRoleAssignment

Adds a role assignment to the user profile.

Syntax

```
IRoleAssignment AddUserRoleAssignment(IHierarchicalZone zone)
IRoleAssignment AddUserRoleAssignment(IHierarchicalZoneComputer computer)
```

Parameters

Specify one of the following parameters when using this method.

zone	The zone to which the Windows user belongs.
computer	The computer to which the Windows user belongs.

Return value

An empty user role assignment object. This role assignment is not stored in Active Directory until you call the RoleAssignment:Commit method.

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, internally Delinea creates a *computer zone* object, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Use the second form of this method to obtain a computer-level role assignment for this user.

Exceptions

AddUserRoleAssignment may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid user or if the method failed to create a role assignment because it cannot find the user.

GetDirectoryEntry

Returns the directory entry for the user from Active Directory.

Syntax

```
DirectoryEntry GetDirectoryEntry()
```

Return value

A directory entry for the service connection point that represents the user's Windows profile.

Discussion

The `DirectoryEntry` object represents the directory object for the user and its associated attributes.

> **Note**: This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

`GetDirectoryEntry` throws an `ApplicationException` if it cannot get the directory object.

Delinea

GetEffectiveUserRoleAssignments

Returns an enumeration of the effective user role assignments.

Syntax

```
IRoleAssignments GetEffectiveUserRoleAssignments()
```

Return value

An enumeration of the effective user role assignments for this user.

Discussion

The collection of effective role assignments is a combination of all the role assignments for this user in this zone and all parent zones. See the Unexpected Link Text class for a more complete discussion.

Name

Gets the Windows login name of the user.

Syntax

```
string Name {get;}
```

Property value

The login name from the user's Windows profile.

WindowsUsers

The WindowsUsers class manages a collection of Windows user objects.

Syntax

```
public interface IWindowsUsers
```

Methods

The WindowsUsers class provides the following method:

Unexpected Link Text	Gets the enumerator you can use to enumerate all Windows user objects.

GetEnumerator

Returns an enumeration of WindowsUser objects.

Syntax

```
IEnumerator GetEnumerator()
```

Return value

Returns an enumerator you can use to list all the WindowsUser objects.

Zone

Manages Delinea zone objects (Centrify.DirectControl.API.IZone).

Syntax

```
public interface IZone
```

Discussion

For each zone you create, you must also define several zone properties. You can also use the Zone class to manage user access rights and the actions users are allowed to perform within a zone. For more information about creating and working with Delinea zones interactively using the Access Manager console, see the *Administrator's Guide for Linux and UNIX*.

Methods

The Zone class provides the following methods:

Unexpected Link Text	Adds an MIT Kerberos realm-trusted user to this zone.
Unexpected Link Text	Commits settings to Active Directory for the zone object.
Unexpected Link Text	Creates a "pending import" group in the zone.
Unexpected Link Text	Creates a "pending import" user in the zone.
Unexpected Link Text	Deletes the zone object from Active Directory.
Unexpected Link Text	Returns the computer profile using the distinguished name (DN) of the profile.
Unexpected Link Text	Returns the list of computers in the zone.
Unexpected Link Text	Returns the directory entry for the Computers parent container object.
Unexpected Link Text	Returns the directory entry for the zone.
Unexpected Link Text	Returns the display name of the zone.
Unexpected Link Text	Returns the directory entry for the Groups parent container object.
Unexpected Link Text	Returns the UNIX group profile for a specified group in the zone.
Unexpected Link Text	Returns the group profile using the distinguished name (DN) of the profile.
Unexpected Link Text	Returns the UNIX group profile for a specified group name in the zone.
Unexpected Link Text	Returns the list of UNIX groups in the zone.
Unexpected Link Text	Returns an individual "pending import" group in the zone.
Unexpected Link Text	Returns the collection of "pending import" groups in the zone.
Unexpected Link Text	Returns an individual "pending import" user in the zone.
Unexpected Link Text	Returns the collection of "pending import" users in the zone.

Unexpected Link Text	Returns the DirectoryEntry of the local groups container.
Unexpected Link Text	Returns the local UNIX group profile for a specified group name in the zone.
Unexpected Link Text	Returns a local group profile using the distinguished name (DN) of the profile.
Unexpected Link Text	Returns the local group profile using the Group Identifier (GID). This method is exposed to the .COM interface.
Unexpected Link Text	Returns a list of the local group profiles in the zone.
Unexpected Link Text	Returns the directory entry of the local users container.
Unexpected Link Text	Returns the local user profile using the specified user name.
Unexpected Link Text	Returns the local user profile specified by the distinguished name (DN) of the profile.
Unexpected Link Text	Returns the local user profile using the User Identifier (UID). This method is exposed to the .COM interface
Unexpected Link Text	Returns a list of the local user profiles in the zone.
Unexpected Link Text	Returns the directory entry for the Users parent container object.
Unexpected Link Text	Returns the user profile using the distinguished name (DN) of the profile.
Unexpected Link Text	Returns the UNIX user profile for a specified user name in the zone.
Unexpected Link Text	Returns the list of UNIX users in the zone.
Unexpected Link Text	Indicates whether a UNIX profile exists for the specified group in the zone.
Unexpected Link Text	Indicates whether a UNIX profile exists in the zone for the specified local group.
Unexpected Link Text	Indicates whether a UNIX profile exists in the zone for the specified local user.
Unexpected Link Text	Adds a computer to the zone.
Unexpected Link Text	Adds a Windows computer to the zone.
Unexpected Link Text	Returns the data stored for the zone object from the data in the Active Directory entry.
Unexpected Link Text	Indicates whether a UNIX profile exists for the specified user in the zone.

Properties

The Zone class provides the following properties:

[AdsiInterfaceadsiinterface.md)	Gets the IADs interface of the zone object in Active Directory.
Unexpected Link Text	Gets the LDAP path to the zone object.
Unexpected Link Text	Gets or sets the Active Directory attribute used for storing the user's password hash.
Unexpected Link Text	Gets or sets the list of available shells for the zone.

Unexpected Link Text	Gets the Cims object managing the zone.
Unexpected Link Text	Gets or sets the default group profile to use as the primary group for new users in the zone.
Unexpected Link Text	Gets or sets the default path to the user's home directory for new users in the zone.
Unexpected Link Text	Gets or sets the default shell assigned to new users in the zone.
Unexpected Link Text	Gets or sets the zone to use for default zone values.
Unexpected Link Text	Gets or sets the description property for the zone.
Unexpected Link Text	Gets the full name of the zone.
Unexpected Link Text	Indicates whether auto-provisioning of group profiles is enabled for the zone.
Unexpected Link Text	Gets the unique identifier for the zone.
Unexpected Link Text	Indicates whether this zone supports hierarchical zone features.
Unexpected Link Text	Indicates whether the zone object's properties are readable.
Unexpected Link Text	Indicates whether the zone uses the Microsoft Services for UNIX (SFU) schema extension.
Unexpected Link Text	Determines whether the zone is a TruncateName zone.
Unexpected Link Text	Indicates whether the zone object's properties are writable.
Unexpected Link Text	Gets or sets the license container associated with this zone.
Unexpected Link Text	Gets or sets the name of the primary domain controller for the zone.
Unexpected Link Text	Determines whether Active Directory group membership must be maintained for UNIX users in the zone.
Unexpected Link Text	Gets or sets the name of the zone.
Unexpected Link Text	Gets or sets the next available GID value for new groups in the zone.
Unexpected Link Text	Gets or sets the next available UID value for new users in the zone.
Unexpected Link Text	Gets or sets the next GID to be used when adding users.
Unexpected Link Text	Gets or sets the next UID to be used when adding users.
Unexpected Link Text	Gets or sets the NIS domain associated with the zone for SFU zones.
Unexpected Link Text	Gets or sets the list of group identifiers (GIDs) that cannot be assigned in the zone.
Unexpected Link Text	Gets or sets the list of User identifiers (UIDs) that cannot be assigned in the zone.
Unexpected Link Text	Gets the schema type of the zone object.
Unexpected Link Text	Gets or sets the Active Directory domain associated with the zone for SFU zones.
Unexpected Link Text	Indicates whether auto-provisioning of user profiles is enabled for the zone.

Unexpected Link Text	Gets the version number of the data schema.

AddMitUser

Adds a UNIX profile for a user in a trusted Kerberos realm to the zone.

Syntax

IUserUnixProfile AddMitUser(string fullMitUserName, int uid, string name, string shell, string homeDir, int primaryGroup)

IUserUnixProfile AddMitUser(string fullMitUserName, long uid, string name, string shell, string homeDir, long primaryGroup)

Parameters

Specify the following parameters when using this method.

fullMitUserName	The full user name of the user and the trusted Kerberos realm. For example: username@mit.realm.name
uid	The value to use as the UID of the user in the specified zone.
name	The UNIX profile name of the user in the specified zone.
shell	The default shell for the user in the specified zone.
homeDir	The default login shell for the user in the specified zone.
primaryGroup	The value to use as the GID for the user's primary group.

Return value

A new UserUnixProfile object.

Discussion

This method creates a UNIX profile object for a user in a trusted realm that is not tied to an Active Directory user object.

There are two versions of this method: one designed for COM-based programs that supports a 32-bit signed number for the uid and primaryGroup arguments and one designed for .NET-based programs that allows a 64-bit signed number for the arguments.

Exceptions

AddMitUser may throw one of the following exceptions:

- ApplicationException if you try to add a Kerberos user to an SFU zone.
- NotSupportedException if the computer zone schema is not supported.

Example

The following code sample illustrates using AddMitUser in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=cohesion_div,
CN=zones,CN=centrify,CN=program data,DC=arcade,DC=com")
'Create the UNIX profile for the Kerberos user “lewis.cain”
set objUserUnixProfile =
objUser.AddMitUser(“lewis.cain@cohesion.org”,98566,”cainl”, “/bin/csh”,
“home/cainl”, 98556)
'Enable the UNIX profile for the new user and update AD
objUserUnixProfile.UnixEnabled = True
objUserUnixProfile.commit
...
```

Delinea

Commit

Commits the settings or changes for a zone object to Active Directory.

Syntax

```
void Commit()
```

Discussion

This method confirms that the zone name and DirectoryEntry attributes are specified before updating Active Directory.

By default, the user who creates the zone object in Active Directory has permission to perform all administrative tasks in the zone. For information about setting and modifying the permissions required to perform specific tasks, see the *Planning and Deployment Guide*.

Exceptions

Commit may throw one of the following exceptions:

- ArgumentException if the zone name is not valid.
- ApplicationException if the container is not a valid zone container, the zone name is already in use, or you have insufficient access rights to modify the zone.
- UnauthorizedAccessException if you have insufficient access rights to commit the data object.
- COMException if an LDAP error occurred. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
'Create a new zone.
set objZone = Cims3.CreateZone(objContainer, strZone)
'set the starting UID and GID for the new zone.
objZone.nextAvailableUID = 10000
objZone.nextAvailableGID = 10000
'set the default shell and home directory the new zone.
objZone.DefaultShell = "/bin/bash"
objZone.DefaultHomeDirectory = "/home/\${user}"
'Finalize the transaction and update Active Directory.
objZone.Commit
...
```

CreateImportPendingGroup

Creates a “pending import” group in the zone.

Syntax

IGroupInfo CreateImportPendingGroup (string source, DateTime timestamp)

Parameters

Specify the following parameters when using this method.

source	String	The location of the source data for the group to be imported.
timestamp	DateTime	The date and time at which the data was retrieved.

Return value

The newly created pending import group object.

Discussion

Group profiles in a “pending import” group object needed to be mapped to Active Directory groups before they can be used. Groups in this state are normally imported from NIS domains or from text files and stored temporarily either in Active Directory or XML files until they are mapped to Active Directory accounts. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Create the Pending Import group object
set objPendGrp = objZone.CreateImportPendingGroup("script file", now)
objPendGrp.Gid = 500
objPendGrp.Name = "users"
objPendGrp.Commit
...
```

CreateImportPendingUser

Creates a “pending import” user in the zone.

Syntax

IUserInfo CreateImportPendingUser(string source, DateTime timestamp)

Parameters

Specify the following parameters when using this method.

source	The location of the source data for the user to be imported.
timestamp	The date and time at which the data was retrieved.

Return value

The newly created pending import user object.

Discussion

User profiles in a pending import user object need to be mapped to Active Directory groups before they can be used. Users in this state are normally imported from NIS domains or from text files and stored temporarily either in Active Directory or in XML files until they are mapped to Active Directory accounts. For more information about importing and mapping users, see the *Administrator's Guide for Linux and UNIX*.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Create the Pending Import User object
set objPendUsr = objZone.CreateImportPendingUser("script file", now)
objPendiUsr.Uid = 500
objPendUsr.Name = "joe"
objPendUsr.PrimaryGroupId = 500
objPendUsr.HomeDirectory = "/home/joe"
objPendUsr.Shell = "/bin/bash"
objPendUsr.Gecos = "Joe Jane"
objPendUsr.Commit
...
```

Delete

Deletes the zone object from Active Directory.

Syntax

```
void Delete()
```

Discussion

To delete a zone, you must have the DeleteSubTree privilege on the zone's parent container. For example, to delete a zone from the default location for new zones, you must have the right to DeleteSubTree allowed on the domain/Program Data/Acme/Zones container object.

Exceptions

Delete may throw one of the following exceptions:

- UnauthorizedAccessException if you have insufficient access rights to delete the zone.
- COMException if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Delete the zone object
objZone.Delete
...
```

GetComputerByDN

Returns the computer profile for a computer in the zone using the distinguished name (DN) of the profile.

Syntax

```
IComputer GetComputerByDN(string dn)
```

Parameter

Specify the following parameter when using this method:

dn	The distinguished name (DN) of the computer profile.

Return value

The computer profile with the distinguished name (DN) matching the distinguished name specified, or null if no matching computer profile is found.

Discussion

The computer profile is the service connection point associated with the computer object.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=default,CN=zones,CN=centrify,CN=program
data,DC=arcade,DC=com")
'Get the computer profile by DN
set objComputer=
objZone.GetComputerByDN("CN=velvet,CN=Computers,CN=default,CN=Zones,
CN=centrify,CN=program data,DC=arcade,DC=com")
wScript.Echo computer.Name
...
```

GetComputers

Returns the list of computers in the zone.

Syntax

IComputers GetComputers()

Return value

The list of computers for the selected zone object.

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=research,CN=zones,CN=centrify,CN=program
data,DC=arcade,
DC=com")
'Display the list of computers in the zone
wScript.Echo “Computers in zone: “
for each computer in objZone.GetComputers
wScript.Echo computer.Name
next
…
```

GetComputersContainer

Returns the parent container object for computer profiles in the zone.

Syntax

```
DirectoryEntry GetComputersContainer()
```

Return value

The DirectoryEntry object of the zone's Computers container.

Discussion

If the Computers container does not exist, this method creates one for you.

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

GetComputersContainer may throw the following exception:

- ApplicationException if you try to use this method in a COM-based program.

GetDirectoryEntry

Returns an instance of the DirectoryEntry object for the zone.

Syntax

```
DirectoryEntry GetDirectoryEntry()
```

Return value

The DirectoryEntry of the zone.

Discussion

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

GetDisplayName

Returns the name displayed for the zone in Access Manager.

Syntax

```
string GetDisplayName()
```

Return value

The DisplayName property for the selected zone object. For example, if using OU=UNIX,OU=Zones for the zone named qa:

```
domain/UNIX/Zones/qa
```

If the zone is defined as a Services for UNIX (SFU) zone, the DisplayName returned ends in SFU. For example:

```
domain/UNIX/Zones/qaSFU
```

Discussion

In most cases, this method returns the same value as the zone. Unexpected Link Text property, for example, domain/UNIX/Zones/testing_lab, and they can be used interchangeably. However, f the zone is defined as a Services for UNIX (SFU) zone supporting the Microsoft Services for UNIX (SFU) schema, this method appends [SFU] to the zone name, for example domain/UNIX/Zones/testing_lab [SFU].

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=research,CN=zones,CN=centrify,CN=program
data,DC=arcade,
DC=com")
'Display the name of the zone
wScript.Echo “Zone name: “ & objZone.GetDisplayName
…
```

GetGroupsContainer

Returns the directory entry for the parent container object for the group profiles in the zone.

Syntax

```
DirectoryEntry GetGroupsContainer()
```

Return value

The directory entry of the zone's Groups container.

Discussion

If the Groups container does not exist, this method creates one for you.

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

GetGroupsContainer throws an ApplicationException if you try to use this method in a COM-based program.

GetGroupUnixProfile

Returns the UNIX group profile for a specified group in the zone.

Syntax

```
IGroupUnixProfile GetGroupUnixProfile(IGroup group)
```

Parameter

Specify the following parameter when using this method:

group	The group for which you want to retrieve profile information.

Return value

The Unexpected Link Text object for the specified group in the zone.

Discussion

This method uses the Centrify.DirectControl.API.IGroup group returned by a Unexpected Link Text or Unexpected Link Text call to retrieve the group profile.

Exceptions

GetGroupUnixProfile may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- NotSupportedException if the zone schema is not supported.
- ApplicationException if there is more than one instance of the specified group in the zone.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Identify the Active Directory group object
set objGrp = cims.GetGroupByPath("LDAP://CN=berlin_qa,CN=Users,DC=ajax,DC=org"))

set objGrpProfile = objZone.GetGroupUnixProfile(objGrp)
'Display the UNIX profile name for the group “berlin_qa”
wScript.Echo objGrpProfile.Name
...
```

GetGroupUnixProfileByDN

Returns the UNIX profile for a group in the zone using the distinguished name (DN) of the profile.

Syntax

IGroupUnixProfile GetGroupUnixProfileByDN(string dn)

Parameter

Specify the following parameters when using this method.

dn	The distinguished name (DN) of the group profile.

Return value

The group profile with the distinguished name (DN) matching the distinguished name specified, or null if no matching group profile is found.

Discussion

The group profile is the service connection point associated with the Active Directory group object.

Exceptions

GetGroupUnixProfile throws a NotSupportedException if the zone schema is not supported.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=default,CN=zones,CN=centrify,CN=program
data,DC=arcade,DC=com")
'Get the group profile by DN
set objComputer= objZone.GetGroupUnixProfileByDN(“CN=legal,CN=Groups,CN=default,
CN=Zones,CN=centrify,CN=program data,DC=arcade,DC=com")
...
```

GetGroupUnixProfileByName

Returns the UNIX group profile for a group with the specified name in the zone.

Syntax

```
IGroupUnixProfile GetGroupUnixProfileByName(string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the UNIX group profile for which you want to retrieve information.

Return value

The GroupUnixProfile object for the specified group name in the selected zone, or null if no group unix profile is found.

Discussion

The name you specify should be the UNIX group name for the group if it differs from the Active Directory name for the group.

Exceptions

GetGroupUnixProfileByName may throw one of the following exceptions:

- NotSupportedException if the zone schema is not supported.
- ApplicationException if there is more than one instance of the specified group in the zone.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Get the UNIX profile for the group “berlin_qa”
set objGrp = objZone.GetGroupUnixProfileByName(“berlin_qa”)
'Display the GID for the group “berlin_qa”
wScript.Echo “GID: “ & objGrp.GID
...
```

GetGroupUnixProfiles

Returns the list of UNIX group profiles that have been defined for the zone.

Syntax

IGroupUnixProfiles GetGroupUnixProfiles()

Return value

The GroupUnixProfiles object for the zone.

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=research,CN=zones,CN=centrify,CN=program
data,DC=arcade,
DC=com")
'Display the list of group profiles defined for the zone
set objGroupProfiles = objZone.GetGroupUnixProfiles
for each objProfile in objGroupProfiles
wScript.Echo “Group profile name: “ & objProfile.Name
next
…
```

GetImportPendingGroup

Returns an individual “pending import” group with the specified ID in the zone.

Syntax

```
IGroupInfo GetImportPendingGroup(string id)
```

Parameter

Specify the following parameter when using this method:

id	The GUID of the “pending import” group profile for which you want to retrieve information.

Return value

The IGroupInfo object for the specified ID in the zone.

Discussion

Group profiles that are “pending import” are normally imported from NIS domains or from text files and not yet mapped to Active Directory groups. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

Example

The following code sample illustrates using this method in a script:

```
...
'Need to obtain an active directory container object
'Configure the test container.
Set objRootDSE = GetObject("LDAP://rootDSE")
set objContainer = GetObject("LDAP://" & strParent & "," &
objRootDSE.Get("defaultNamingContext"))
strContainerDN = objContainer.get("DistinguishedName")
'Get the zone object.
Set objZone = cims.GetZoneByPath("cn=" & strZone & "," & strContainerDN)
Set objGroupInfo = objZone.GetImportPendingGroup(strID)
objGroupInfo.Delete
...
```

GetImportPendingGroups

Returns the list of “pending import” groups for the zone.

Syntax

```
IGroupInfos GetImportPendingGroups()
```

Return value

The collection of “pending import” group profiles for the zone.

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=default,CN=zones,CN=centrify,CN=program data,
DC=arcade,DC=com")
'Get the collection of pending import groups
set objPendingGrps = objZone.GetImportPendingGroups
if not objPendingGrps is nothing then
wScript.Echo "Pending import groups: ", objPendingGrps.Count
for each objPendingGrp in objPendingGrps
wScript.Echo objPendingGrp.Gid, objPendingGrp.Name
if objPendingGrp.Source = "script file" then
objPendingGrp.Delete
end if
next
wScript.Echo ""
end if
…
```

GetImportPendingUser

Returns an individual “pending import” user with the specified ID in the zone.

Syntax

```
IUserInfo GetImportPendingUser(string id)
```

Parameter

Specify the following parameter when using this method:

id	The GUID of the “pending import” user profile for which you want to retrieve information.

Return value

The Unexpected Link Text object for the specified ID in the zone.

Discussion

User profiles that are “pending import” are normally imported from NIS domains or from text files and not yet mapped to Active Directory users. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

Example

The following code sample illustrates using this method in a script:

```
...
// Get the user object
IUser objUser = cims.GetUserByPath(strUser);
// Get the zone object
IZone objZone = cims.GetZoneByPath("cn=" + strZone + "," + strContainerDN);
Import.IUserInfo objUserInfo = objZone.GetImportPendingUser(strUserInfo);
if (objUser == null)
{
    Console.WriteLine("User " + strUser + " does not exist.");
}
else
{
    objUserInfo.Import(objUser);
}
...
```

GetImportPendingUsers

Returns the list of “pending import” users for the zone.

Syntax

```
IUserInfos GetImportPendingUsers()
```

Return value

The collection of “pending import” user profiles for the zone.

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
‘Get the collection of pending users
set objPendingUsrs = objZone.GetImportPendingUsers
if not objPendingUsrs is nothing then
wScript.Echo "Pending import users: ", objPendingUsrs.Count
for each objPendingUsr in objPendingUsrs
wScript.Echo objPendingUsr.Uid, objPendingUsr.Name
if objPendingUsr.Source = "script file" then
objPendingUsr.Delete
end if
next
wScript.Echo ""
end if
…
```

GetLocalGroupsContainer

Returns the directory entry for the parent container object for the local group profiles.

Syntax

```
DirectoryEntry GetLocalGroupsContainer()
```

Return value

The directory entry of the local group's container.

Discussion

This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for `directory` objects. This method cannot be used in COM-based programs.

Exceptions

`GetLocalGroupsContainer` may throw the following exception:

`ApplicationException` if you try to use this method in a COM-based program.

GetLocalGroupUnixProfile

Returns the UNIX group profile for a specified local group.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfile(string groupName)
```

Parameter

Specify the following parameter when using this method:

groupName	The name of the local group for which you want to retrieve profile information.

Return value

The Unexpected Link Text object for the specified local group name. If there is no group, `null` is returned.

Exceptions

`GetGroupUnixProfile` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is null.

GetLocalGroupUnixProfileByDN

Returns the Local UNIX profile for a group in the zone using the distinguished name (DN) of the profile.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfileByDN(string dn)
```

Parameter

Specify the following parameters when using this method.

dn	The distinguished name (DN) of the local group profile.

Return value

The local group profile with the distinguished name (DN) matching the distinguished name specified, or null if no matching group profile is found.

GetLocalGroupUnixProfileByGid (Int32)

Returns the Local UNIX profile for a group in the zone using the group identifier (GID) of the profile. This method is exposed to the .COM interface.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfileByGid(int gid)
```

Parameter

Specify the following parameters when using this method.

gid	The group identifier (GID) of the local group profile.

Return value

The local group profile with the specified group identifier (GID) or `null` if no matching group profile is found.

GetLocalGroupUnixProfiles

Get the list of local group profiles in the zone.

Syntax

```
IGroupUnixProfiles GetLocalGroupUnixProfiles()
```

Return value

Returns a collection of `GroupUnixProfile` objects. If there are no groups, `null` is returned.

GetLocalUsersContainer

Returns the directory entry for the parent container object for the local user profiles in the zone.

Syntax

```
DirectoryEntry GetLocalUsersContainer()
```

Return value

The directory entry of the zone's users container.

Discussion

If the `Users` container does not exist, this method creates one for you.

This method can only be used in .NET programs because `DirectoryEntry` is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

`GetUsersContainer` may throw the following exception:

- `ApplicationException` if you try to use this method in a COM-based program.

GetLocalUserUnixProfile

Returns the UNIX user profile for a specified local group.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfile(string userName)
```

Parameter

Specify the `userName` parameter when using this method.

Return value

Returns the local user profile with the specified user name. If there is no group, `null` is returned.

GetLocalUserUnixProfileByDN

Returns the local UNIX profile for a user in the zone using the distinguished name (DN) of the profile.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfileByDN(string dn)
```

Parameter

Specify the following parameters when using this method.

dn	The distinguished name (DN) of the local user profile.

Return value

Returns the local user profile with the distinguished name (DN) matching the distinguished name specified, or null if no matching group profile is found.

GetLocalUserUnixProfileByUid (Int32)

Returns the local UNIX profile for a user in the zone using the user identifier (GID) of the profile. This method is exposed to the .COM interface.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfileByUid(int uid)
```

Parameter

Specify the following parameters when using this method.

uid	The user identifier (UID) of the local user profile.

Return value

The local user profile with the specified user identifier (UID) or null if no matching user profile is found.

GetLocalUserUnixProfiles

Get a list of local UNIX user profiles in the zone.

Syntax

```
IUserUnixProfiles GetLocalUserUnixProfiles()
```

Return value

Returns a collection of local user profiles in the zone. If there are no users, `null` is returned.

GetUsersContainer

Returns the directory entry for the parent container object for the user profiles in the zone.

Syntax

```
DirectoryEntry GetUsersContainer()
```

Return value

The directory entry of the zone's Users container.

Discussion

If the Users container does not exist, this method creates one for you.

This method can only be used in .NET programs because DirectoryEntry is a .NET-specific class for directory objects. This method cannot be used in COM-based programs.

Exceptions

`GetUsersContainer` may throw the following exception:

- `ApplicationException` if you try to use this method in a COM-based program.

GetUserUnixProfileByDN

Returns the UNIX profile for a user in the zone using the distinguished name (DN) of the profile.

Syntax

IUserUnixProfile GetUserUnixProfileByDN(string dn)

Parameter

Specify the following parameter when using this method:

dn	The distinguished name (DN) of the user profile.

Return value

The user profile with the distinguished name (DN) matching the distinguished name specified, or null if no matching user profile is found.

Discussion

The user profile is the service connection point associated with the Active Directory user object.

Exceptions

GetUserUnixProfileByDN throws a NotSupportedException if the zone schema is not supported.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone =
cims.GetZoneByPath("LDAP://CN=default,CN=zones,CN=centrify,CN=program
data,DC=arcade,DC=com")
'Get the user profile by DN
set objComputer=
objZone.GetUserUnixProfileByDN("CN=yuji,CN=Users,CN=default,CN=Zones,CN=centrify,CN=program
data,DC=arcade,DC=com")
...
```

GetUserUnixProfileByName

Returns the UNIX user profile associated with the specified user name in the zone.

Syntax

```
IUserUnixProfile GetUserUnixProfileByName(string name)
```

Parameter

Specify the following parameter when using this method:

name	The user's UNIX login name.

Return value

The UserUnixProfile object associated with the specified user name in the zone, or null if the UserUnixProfile is not found.

Exceptions

GetUserUnixProfileByName may throw one of the following exceptions:

- NotSupportedException if the zone schema is not supported.
- ApplicationException if there is more than one instance of the specified user in the zone.

Example

The following code sample illustrates using this method in a script:

```
…
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Display the UNIX profile for the group “jae”
wScript.Echo objZone.GetUserUnixProfileByName(“jae”)
…
```

GetUserUnixProfiles

Returns the list of UNIX user profiles for the zone.

Syntax

IUserUnixProfiles GetUserUnixProfiles()

Return value

The Unexpected Link Text object for the zone. This object is a collection of UserUnixProfile objects.

Example

The following code sample illustrates using this method in a script to enumerate all of the user profiles in the default zone:

```
...
'Specify the zone you want to work with
Set objZone = cims.GetZone("acme.com/Program Data/Acme/Zones/default")
'List the UNIX login name for each profile in the zone
Set objProfiles = objZone.GetUserUnixProfiles()
For each profile in objProfiles
wscript.echo profile.Name
next
...
```

GroupUnixProfileExists

Checks whether a UNIX profile exists for the specified group in the zone.

Syntax

```
bool GroupUnixProfileExists(IGroup group)
```

Parameter

Specify the following parameter when using this method:

group	The group name for which you want to check whether a UNIX profile exists.

Return value

Returns true if a UNIX profile is found in the zone for the specified group, or false if no UNIX profile exists for the group in the zone.

Exceptions

GroupUnixProfileExists may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- NotSupportedException if the zone schema is not supported.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Check whether there's a UNIX profile for the group "legal"
if objZone.GroupUnixProfileExists(legal) = true
wScript.Echo "Profile exists in this zone"
else
wScript.Echo "No matching profile in this zone!"
end if
...
```

LocalGroupUnixProfileExists

Checks whether a UNIX profile exists for the specified local group in the zone.

Syntax

```
bool LocalGroupUnixProfileExists(string groupName)
```

Parameter

Specify the following parameter when using this method:

groupName	The group name for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if the local UNIX group profile is found in the zone, or `false` if no UNIX profile exists for the group in the zone.

Exceptions

`LocalGroupUnixProfileExists` may throw the following exception:

- `ArgumentNullException` if the specified parameter value is null.

LocalUserUnixProfileExists

Checks whether a local UNIX profile exists for the specified user in the zone.

Syntax

```
bool LocalUserUnixProfileExists(string userName)
```

Parameter

Specify the following parameter when using this method:

userName	The local user name for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if a UNIX profile is found in the zone for the specified local user, or `false` if no UNIX profile exists for the user in the zone.

Exceptions

`UserUnixProfileExists` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is null.

PrecreateComputer

Adds a computer to a zone.

Syntax

IComputer PrecreateComputer(DirectoryEntry adComputerEntry, string[] spn, DirectoryEntry trustee)

IComputer PrecreateComputer(DirectoryEntry containerEntry, string cn, string dnsName, string[] spn, DirectoryEntry trustee

IComputer PrecreateComputer(DirectoryEntry adComputerEntry, string[] spn, DirectoryEntry trustee, bool skipPermissionSetting);

IComputer PrecreateComputer(DirectoryEntry containerEntry, string cn, string dnsName, string[] spn, DirectoryEntry trustee, bool skipPermissionSetting);

Parameters

Specify the following parameters when using this method.

adComputerEntry	The DNS host name of the Active Directory computer object you wish to add to the zone.
containerEntry	The Directory container for the created computer.
cn	The computer name.
dnsName	The DNS name of the created computer.
skipPermissionSetting	Specifies if permission delegation is skipped when precreating computers.
spn	Service Principal Name. Specify null to use default.
trustee	The user or group to delegate adjoin permissions to, Specify null to delegate the permission for a self-service join.
trusteeDn	The user or group to which the computer-level overrides will be assigned, specified as a distinguished name.

Return value

The computer object that is added to the zone.

Discussion

Use PrecreateComputer(DirectoryEntry, string[], DirectoryEntry) to add an existing Active Directory computer to the zone. Use PrecreateCompute(DirectoryEntry, string, string, string[], DirectoryEntry) to create a new UNIX computer object and add it to the zone.

PrecreateWindowsComputer

Adds an existing Windows computer to a zone.

Syntax

```
IComputer PrecreateWindowsComputer(DirectoryEntry adComputerEntry);
IComputer PrecreateWindowsComputer(DirectoryEntry adComputerEntry, bool skipPermissionSetting);
```

Parameters

Specify the following parameter when using this method.

adComputerEntry	The DNS host name of the Windows computer object you wish to add to the zone.
skipPermissionSetting	Specifies if permission delegation is skipped when precreating computers.

Return value

The computer object that is added to the zone.

Refresh

Reloads the zone object data from the data in Active Directory.

Syntax

```
void Refresh()
```

Discussion

This method refreshes the zone information in the cached object to ensure it is synchronized with the latest information in Active Directory.

Exceptions

`Refresh` may throw the following exception:

- `COMException` if an LDAP error occurs. LDAP errors can occur if the connection to the LDAP server fails, the connection times out, invalid credentials are presented, or there are other problems communicating with Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZoneByPath("LDAP://CN=corporate,CN=zones,CN=centrify,
CN=program data,DC=sierra,DC=com")
'Change the zone description
objZone.Description = "Corporate offices, Edinburgh”
objZone.Commit
'Reload the zone object
objZone.Refresh
...
```

UserUnixProfileExists

Checks whether a UNIX profile exists for the specified user in the zone.

Syntax

```
bool UserUnixProfileExists(IUser user)
```

Parameter

Specify the following parameter when using this method:

user	The user name for which you want to check whether a UNIX profile exists.

Return value

Returns true if a UNIX profile is found in the zone for the specified user, or false if no UNIX profile exists for the user in the zone.

Exceptions

UserUnixProfileExists may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- NotSupportedException if the zone schema is not supported.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Check whether there's a UNIX profile for the user "garcia"
if objZone.GroupUnixProfileExists(garcia) = true
wScript.Echo "Profile exists in this zone"
else
wScript.Echo "No matching profile in this zone!"
end if
...
```

AdsiInterface

Gets the IADs interface of the zone object from Active Directory.

Syntax

IADs AdsiInterface {get;}

Property value

The IADs interface of the zone object.

Discussion

This property enables you to perform any operations provided by the underlying Active Directory Service Interfaces (ADSI) for a zone as a directory object. For example, you can use this property to retrieve the IADs properties and methods that enable you to access the security information for the zone object.

Example

The following code sample illustrates using AdsInterface in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab,cn=Zones,cn=UNIX,dc=ajax,dc=org")
'Get the IADs for the zone
set secdes = zone.AdsiInterface.Get("ntSecurityDescriptor")
'Display security information for the zone
wScript.Echo secdes.Owner
...
```

ADsPath

Gets the LDAP path to the zone object.

Syntax

```
string ADsPath {get;}
```

Property value

The full LDAP path to the zone object.

Example

The following code sample illustrates using ADsPath in a script:

```
...
'Specify the zone you want to work with
set zone = GetZone("fireball.net/Field/Zones/macs")
'Display the LDAP path for the zone
wScript.Echo zone.ADsPath
...
```

AgentlessAttribute

Gets or sets the Active Directory attribute used for storing the user's password hash if a zone supports agentless NIS client requests.

Syntax

```
string AgentlessAttribute {get; set;}
```

Property value

The Active Directory attribute used for storing the user's password hash.

Discussion

If you have any computers configured to respond to NIS client requests using information stored in Active Directory, this property must be set to enable password synchronization for the zone. Only the following values are valid:

- altSecurityIdentities
- msSFU30Password
- unixUserPassword

Setting this property to an invalid value disables password synchronization.

Exceptions

AgentlessAttribute throws an ApplicationException if the selected attribute cannot store a password hash.

Example

The following code sample illustrates setting this property in a script:

```
...
'Specify the zone you want to work with
set zone = cims.GetZone("zap.org/Program Data/Acme/Zones/default")
'Change the attribute used for the password hash
zone.AgentlessAttribute = "unixUserPassword"
zone.Commit()
...
```

AvailableShells

Gets or sets the list of available shells for this zone.

Syntax

```
string[ ] AvailableShells {get; set;}
```

Property value

The list of available shells for the zone.

Discussion

The values you define for this property are used as the values in the drop-down list of available shells when defining the UNIX profile for a new user in the Access Manager console.

This property requires a strongly-typed array. Because strongly-typed arrays are not supported in VBScript, you cannot use this property in scripts written with VBScript. To use this property, you must use a programming language that allows strongly-typed arrays.

Example

The following code sample illustrates setting this property in a Visual Studio (C#) script:

```
...
// Set the Active Directory container object.
DirectoryEntry objContainer = new
DirectoryEntry("LDAP://cn=Zones,cn=UNIX,dc=ajax,dc=org");
IZone objZone = cims.CreateZone(objContainer, “QA Zone”);

// set the starting UID and GID for the zone
objZone.NextAvailableUID = 10000;
objZone.NextAvailableGID = 10000;

// set the list of available shells and default shell for the zone
objZone.AvailableShells = new string[] {"/bin/bash", "/bin/shell"};
objZone.DefaultShell = "/bin/bash";

// set the default home directory for the zone
objZone.DefaultHomeDirectory = "/home/\${user}";
objZone.Commit();
Console.WriteLine("Zone created successfully.");
...
```

Cims

Gets the Cims object managing the zone.

Syntax

```
Cims Cims {get;}
```

Property value

The Cims object managing this zone.

Discussion

This property serves as a shortcut for retrieving data.

DefaultGroup

Gets or sets the default group profile to use as the primary group for new users in the zone.

Syntax

```
IGroupUnixProfile DefaultGroup{get; set;}
```

Property value

The default group property for the zone.

Discussion

The default group profile for a zone is always associated with an existing Active Directory group. You can, however, define a primary group that is not associated with any Active Directory group.

For more information about defining primary groups for UNIX users, see the *Planning and Deployment Guide* and the *Administrator's Guide for Linux and UNIX*.

Example

The following code sample illustrates using `DefaultGroup` in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Check the default group for the zone
If objZone.DefaultGroup is nothing
'Identify the Active Directory group object
set objGrp = cims.GetGroupByPath("LDAP://CN=IT Interns,CN=Users,DC=ajax,DC=org")

'Get the UNIX profile for the Active Directory group
set objGrpProfile = objZone.GetGroupUnixProfile(objGrp)
'Make this profile the default group
set objZone.DefaultGroup = objGrpProfile
end if
objZone.Commit
...
```

DefaultHomeDirectory

Gets or sets the local file system path to the user's default home directory.

Syntax

```
string DefaultHomeDirectory {get; set;}
```

Property value

The text string that defines the default path to the user's home directory.

Discussion

The only variable permitted is $. The Access Manager console replaces this variable with the user's UNIX login name when you add a UNIX profile for the user to the zone.

Example

The following code sample illustrates using DefaultHomeDirectory in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("zap.org/Program Data/Acme/Zones/default")
'Set zone properties
objZone.DefaultHomeDirectory = "/home/\${user}"
objZone.DefaultShell = "/bin/bash"
objZone.NextAvailableUID = zone.NextAvailableUID + 1
objZone.NextAvailableGID = zone.NextAvailableGID + 1
objZone.DefaultHomeDirectory = "/home/\${user}"
...
```

DefaultShell

Gets or sets the default shell assigned to new users in the zone.

Syntax

```
string DefaultShell {get; set;}
```

Property value

The default shell property for the zone.

Discussion

The value you define for this property is automatically populated as the default shell when defining the UNIX profile for a new user in Access Manager. The value can be overridden by the administrator defining the user's profile.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("zap.org/Program Data/Acme/Zones/default")
'Specify zone properties
objZone.DefaultHomeDirectory = "/home/admin"
objZone.DefaultShell = "/bin/bash"
objZone.NextAvailableUID = zone.NextAvailableUID + 1
objZone.NextAvailableGID = zone.NextAvailableGID + 1
...
```

DefaultValueZone

Gets or sets the zone to use as the “master” zone for setting default zone property values.

Syntax

IZone DefaultValueZone {get; set;}

Property value

The zone object for the zone used to define default values.

Discussion

If this property is set, the profile information and zone properties in the specified zone are used as the default values for the current zone. For example, if you add users or groups that have profiles in the specified zone to the zone context in which you are currently working, their UNIX profiles have the same UIDs and GIDs in both zones by default.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Get the master default values zone for this zone
set objMaster = objZone.DefaultValueZone
wScript.Echo “The master zone is “ & objMaster.Name
...
```

Description

Gets or sets the text string used for the description property of the zone.

Syntax

```
string Description {get; set;}
```

Property value

The description property for the zone.

Discussion

The zone description can consist of any text string up to the number maximum of characters available in the data attribute where zone properties are stored. The maximum number of characters available for the attribute is approximately 850, but the maximum available for the description depends on the other data being stored. For example, the more available shells you have defined for a zone, the shorter the zone description must be.

Example

The following code sample illustrates setting this property in a script:

```
...
'Specify the zone you want to work with
set zone = cims.GetZone("fireball.net/Field/Zones/macs")
'Set the long description for the zone
zone.Description = "Mac OS X computers and users in Fireball field offices"
zone.Commit()
...
```

FullName

Gets the full name of the zone.

Syntax

```
string FullName {get;}
```

Property value

The full, canonical name for the zone.

Discussion

The full name is the canonical name of the zone. The full name is updated when the directory object is updated. Therefore, changes to the Unexpected Link Text property are not reflected in the value retrieved by the FullName property until the changes to the Name property are committed to Active Directory.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Display the full name zone
If objZone.IsReadable = true
wScript.Echo "Zone name: " & objZone.FullName
end if
...
```

GroupAutoProvisioningEnabled

Indicates whether auto-provisioning of group profiles is enabled for the zone.

Syntax

```
bool GroupAutoProvisioningEnabled {get;}
```

Property value

Returns `true` if the zone has auto-provisioning enabled.

Discussion

When automatic provisioning is enabled for groups, the Zone Provisioning Agent can automatically provision new UNIX profiles for groups added to the zone. In addition to enabling provisioning, you must specify a provisioning group to use as the source for provisioning data. For more information about automatic provisioning of users and groups, see the *Planning and Deployment Guide*.

ID

Gets the unique identifier for the zone.

Syntax

```
string ID {get;}
```

Property value

The unique identifier for the zone.

Discussion

This property is used internally to prevent a zone from being listed more than once.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab,cn=Zones,cn=UNIX,dc=ajax,dc=org")
'Display the unique identifier for the zone
wScript.Echo zone.ID
...
```

IsHierarchical

Indicates whether the zone supports hierarchical zone features.

Syntax

```
bool IsHierarchical {get;}
```

Property value

Returns true if the zone is hierarchical.

IsReadable

Determines whether this zone object's properties are readable for the user whose credentials are presented.

Syntax

```
bool IsReadable {get;}
```

Property value

Returns true if the zone object is readable by the user, or false if the zone object is not readable.

Discussion

This property returns a value of true if the user accessing the zone object in Active Directory has sufficient permissions to read zone properties.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Check whether the zone is readable
If not objZone.IsReadable then
wScript.Echo “You do not have read permission for this zone”
end if
...
```

IsSFU

Determines whether the zone uses the Microsoft Services for UNIX (SFU) schema extension.

Syntax

```
bool IsSFU {get;}
```

Property value

Returns true if the zone uses a Services for UNIX (SFU) schema, or false if the zone does not use the SFU schema.

Discussion

If the Microsoft Services for UNIX (SFU) schema extension is installed and being used to store UNIX attributes for the zone, this property returns a value of true.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab,cn=Zones,cn=UNIX,dc=ajax,dc=org")
'Check whether the zone uses the SFU schema
If zone.IsSFU then
wScript.Echo “Zone uses the SFU schema for UNIX attributes”
end if
...
```

IsTruncateName

Determines whether the zone is a TruncateName zone.

Syntax

```
bool IsTruncateName {get; set;}
```

Property value

If this property is true, the zone is a TruncateName zone.

Discussion

If this property is true, the default pre-Windows 2000 name for new users is the computer account name truncated at 15 characters.

IsWritable

Determines whether the zone object's properties are writable properties for the user whose credentials are presented.

Syntax

```
bool isWritable {get;}
```

Property value

Returns true if the zone object is writable by the user, or false if the zone object is not writable.

Discussion

This property returns a value of true if the user accessing the zone object in Active Directory has sufficient permissions to change zone properties.

Example

The following code sample illustrates using this property in a script:

```
…
'Specify the zone you want to work with
set objZone = cims.GetZone("ajax.org/UNIX/Zones/test_lab")
'Check whether the zone is writable
If not objZone.IsWritable then
wScript.Echo “You do not have permission to modify the zone”
end if
…
```

Licenses

Gets or sets the license container associated with this zone.

Syntax

```
ILicenses Licenses {get; set;}
```

Property value

The license container for this zone.

MasterDomainController

Gets or sets the name of the domain controller to use as the primary domain controller for the zone.

Syntax

```
string MasterDomainController {get; set;}
```

Property value

The name of the primary domain controller for the zone.

Example

The following code sample illustrates using MasterDomainController in a script:

```
…
'Specify the zone you want to work with
set zone = GetZone("fireball.net/Field/Zones/macs")
‘Display the domain controller for this zone
wScript.Echo “Main Domain Controller: “ & zone.MasterDomainController
…
```

MustMaintainADGroupMembership

Gets or sets the flag that indicates whether Active Directory group membership must be maintained.

Syntax

```
bool MustMaintainADGroupMembership {get; set;}
```

Property value

Returns true if the Active Directory group membership must be maintained; otherwise, it returns false.

Discussion

By default, setting the primary Active Directory group in a user's UNIX profile does not affect the user's actual Active Directory group membership.

If you want to enforce Active Directory group membership for new users when you add them to the zone, set this property to true. Setting this property to true displays the **Associate Active Directory group membership** option in the Zone Properties dialog box.

Example

The following code sample illustrates using MustMaintainADGroupMembership in a script:

```
...
'Specify the zone you want to work with
set zone = GetZone("fireball.net/Field/Zones/macs")
'Check whether Active Directory group membership is enforced
if zone.MustMaintainADGroupMembership then
wScript.Echo "Active Directory group membership maintained"
else
wScript.Echo "No Active Directory group membership needed"
end if
...
```

Name

Gets or sets the name of the zone.

Syntax

```
string name {get; set;}
```

Property value

The short name of the zone. The name must start with an alphanumeric character or an underscore (*) character and can contain any combination of letters (upper- or lowercase), numerals 0 through 9, and the period (.), hyphen (-) and underscore (*) characters up to a maximum length of 64 characters.

Exceptions

Name throw an ArgumentException if you try to set an invalid name for the zone.

Example

The following code sample illustrates setting this property in a script:

```
…
'Specify the zone you want to work with
set zone = cims.GetZone("zap.org/Program Data/Acme/Zones/default")
'Change the name of the “default” zone to “Pilot deployment”
zone.Name = "Pilot deployment"
zone.Commit()
…
```

NextAvailableGID

Returns or sets the next available value for the group identifier (GID) in the zone.

Syntax

```
int NextAvailableGID {get; set;}
```

Property value

The numeric value of the next available GID for the zone.

Discussion

This method returns or sets the next available GID to be used as the default GID assignment for the next group given access to the zone. If you are setting this property as part of creating a new zone, use this value to define the starting GID value for all groups in the zone. In most cases, this value is incremented automatically each time a new group profile is created for the zone. If you are creating new groups programmatically, use this property to read the current value.

There are two versions of this property: one designed for COM-based programs (NextAvailableGID) that supports a 32-bit signed number for the GID and one designed for .NET-based programs (Unexpected Link Text) that allows a 64-bit signed number for the GID. You can use either property.

Example

The following code sample illustrates setting this property in a script:

```
...
'Set the container object for the zone
set objContainer = GetObject("LDAP://cn=Zones,cn=UNIX, dc=ajax,dc=org")
'Create a new zone named “Sample_Zone”
set objZone = cims.CreateZone(objContainer, “Sample_Zone”)
'Set the starting UID and GID for the new zone
objZone.nextAvailableUID = 10000
objZone.nextAvailableGID = 10000
...
```

NextAvailableUID

Returns or sets the next available value for the user identifier (UID) in the zone.

Syntax

```
int nextAvailableUID {get; set;}
```

Property value

The numeric value of the next available UID for the zone.

Discussion

This method returns or sets the next available UID to be used as the default UID assignment for the next user given access the zone. If you are setting this property as part of creating a new zone, use this value to define the starting UID for all users in the zone. In most cases, this value is incremented automatically each time a new user is enabled for the zone. If you are creating new users programmatically, you can use this property to read the current value.

There are two versions of this property: one designed for COM-based programs (nextAvailableUID) that supports a 32-bit signed number for the UID and one designed for .NET-based programs (Unexpected Link Text) that allows a 64-bit signed number for the UID. You can use either method.

Example

The following code sample illustrates setting this property in a script:

```
...
'Specify the zone you want to work with
set objZone = cdc.GetZone("ajax.org/UNIX/Zones/ea_central")
'Reset the next available UID for the zone
objZone.nextAvailableUID = 5000
zone.Commit()
...
```

NextGID

Gets or sets the next GID to be used when adding groups (64-bit for use with .NET).

Syntax

```
long NextGID {get; set;}
```

Property value

The GID for new groups.

Discussion

This method returns or sets the next available GID to be used as the default GID assignment for the next group given access to the zone. If you are setting this property as part of creating a new zone, use this value to define the starting GID value for all groups in the zone. In most cases, this value is incremented automatically each time a new group profile is created for the zone. If you are creating new groups programmatically, use this property to read the current value.

There are two versions of this property: one designed for COM-based programs (Unexpected Link Text) that supports a 32-bit signed number for the GID and one designed for .NET-based programs (NextGID) that allows a 64-bit signed number for the GID. You can use either method.

NextUID

Gets or sets the next UID to be used when adding users (64-bit for use with .NET).

Syntax

```
long NextUID {get; set;}
```

Property value

The UID for new users.

Discussion

This method returns or sets the next available UID to be used as the default UID assignment for the next user given access to the zone. If you are setting this property as part of creating a new zone, use this value to define the starting UID for all users in the zone. In most cases, this value is incremented automatically each time a new user is enabled for the zone. If you are creating new users programmatically, you can use this property to read the current value.

There are two versions of this property: one designed for COM-based programs (NextAvailableGID](nextavailablegid.md)) that supports a 32-bit signed number for the UID and one designed for .NET-based programs (NextUID) that allows a 64-bit signed number for the UID. You can use either method.

NISDomain

Gets or sets the NIS domain associated with the zone when the zone is determined to be a zone that uses the Microsoft Services for UNIX (SFU) schema extension or is configured to support agentless NIS client requests.

Syntax

```
string NISDomain {get; set;}
```

Property value

The Network Information Service (NIS) distinguished name for the zone.

Discussion

If the zone is a Services for UNIX (SFU) zone, this property should be the NIS domain defined in users' UNIX attributes. For agentless client requests, the zone associated with the computer acting as the NIS server is the NIS domain.

Exceptions

NISDomain throws an ApplicationException if no value is specified when setting this property. You must specify a value when using this property to set the NIS domain.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab,cn=Zones,cn=UNIX,dc=ajax,dc=org")
'If the zone uses the SFU schema, display its NIS domain
If zone.IsSFU then
wScript.Echo "NIS Domain: " & zone.NISDomain
end if
...
```

ReservedGID

Gets or sets the list of reserved group identifiers (GIDs) in the zone.

Syntax

```
string[ ] ReservedGID {get; set;}
```

Discussion

Reserved GIDs cannot be assigned when creating new groups. The `get` argument returns a string containing the range of GIDs not available. The `set` argument specifies a number range to be reserved.

This property requires a strongly-typed array. Because strongly-typed arrays are not supported in VBScript, you cannot use this property in scripts written with VBScript. To use this property, you must use a programming language that allows strongly-typed arrays.

Example

The following code sample illustrates using `ReservedUID` in a Visual Studio (C#) script:

```
...
IZone objZone = cims.CreateZone(objContainer, strZone);
// Set the starting UID and GID for the zone
objZone.NextAvailableUID = 10000;
objZone.NextAvailableGID = 10000;

// Set the reserved UIDs and GIDs for the zone
objZone.ReservedUID = new string[] {"0-300", "999"};
objZone.ReservedGID = new string[] {"0-300", "999"};
objZone.Commit();
...
```

ReservedUID

Gets or sets the list of reserved user identifiers (UIDs).

Syntax

```
string[ ] ReservedUID {get; set;}
```

Discussion

Reserved UIDs cannot be assigned when creating new users. The get argument returns a string containing the range of UIDs not available. The set argument specifies a number range to be reserved.

This property requires a strongly-typed array. Because strongly-typed arrays are not supported in VBScript, you cannot use this property in scripts written with VBScript. To use this property, you must use a programming language that allows strongly-typed arrays.

Example

The following code sample illustrates using ReservedUID in a Visual Studio (C#) script:

```
...
IZone objZone = cims.CreateZone(objContainer, strZone);
// Set the starting UID and GID for the zone
objZone.NextAvailableUID = 10000;
objZone.NextAvailableGID = 10000;

// Set the reserved UIDs and GIDs for the zone
objZone.ReservedUID = new string[] {"0-300", "999"};
objZone.ReservedGID = new string[] {"0-300", "999"};
objZone.Commit();
...
```

Schema

Gets the schema type of the zone object.

Syntax

```
ZoneSchema Schema {get;}
```

Property value

The schema type for the zone.

Discussion

The schema type defines how data for the zone should be stored in Active Directory and is based on the specific Active Directory schema you are using. Zones can be defined as:

- Standard Delinea zones
- Standard Delinea RFC 2307-compliant zones
- Delinea Services for UNIX (SFU) zones

The schema type provides an additional level of granularity corresponding the specific version of the Active Directory schema you are using and where specific zone properties and UNIX attributes are stored. The schema types currently defined for Centify zones are:

Unknown	-1	Schema unknown
Dynamic_Schema_1_0	0	Standard Delinea zone, version 1.x Uses the Delinea version 1.x and standard Active Directory schema data storage model. This zone type is for backward compatibility and otherwise no longer in use.
Dynamic_Schema_2_0	1	Standard Delinea zone, version 2.x and 3.x Uses the Delinea version 2.x and standard Active Directory schema data storage model. This zone type is for backward compatibility and otherwise no longer in use.
SFU_3_0	2	SFU zone, version 2.x and 3.x Uses a combination of the Delinea version 3.x and Microsoft Services for UNIX (SFU) 3.0 data storage model. This zone type can be used when Active Directory has the Microsoft Services for UNIX (SFU), version 3.x, schema extension installed. The standard UNIX properties are stored as defined by the Microsoft SFU 3.x schema, but associated with zones. This zone type is for backward compatibility if you have the Microsoft Services for UNIX (SFU) schema extension installed, and otherwise no longer in use.
SFU_4_0	3	SFU zone, version 4.x Uses a combination of the Delinea version 3.x and Microsoft Services for UNIX (SFU) 4.0 data storage model. This zone type can be used when Active Directory has the Microsoft Services for UNIX (SFU), version 4.0, schema extension installed. The standard UNIX properties are stored as defined by the Microsoft SFU 4.0 schema, but associated with zones. This zone type is for backward compatibility if you have the Microsoft Services for UNIX (SFU) schema extension installed, and otherwise no longer in use.
CDC_RFC_2307	5	Standard RFC 2307-compatible zone, version 3.x Uses the Active Directory RFC 2307-compliant schema data storage model.
Dynamic_Schema_3_0	6	Standard Delinea zone, version 3.x and 4.x Uses the Delinea version 4.x and Active Directory schema data storage model. **Note**: The only difference between the Dynamic_Schema_2_0 data storage model and the Dynamic_Schema_3_0 data storage model is the use of the managedBy attribute. This attribute is set in zones that use the Dynamic_Schema_2_0 schema. The managedBy attribute is not used in zones that use in the Dynamic_Schema_3_0 schema.
CDC_RFC_2307_2	7	Classic RFC 2307-compatible zone, version 4.x Uses the Active Directory RFC 2307-compliant schema data storage model. **Note**: The only difference between the CDC_RFC_2307 data storage model and the CDC_RFC_2307_2 data storage model is the use of the managedBy attribute. This attribute is set in zones that use the CDC_RFC_2307 schema. The managedBy attribute is not used in zones that use in the CDC_RFC_2307_2 schema.

Dynamic_Schema_5_0	8	Hierarchical zone, version 5.x Uses the Delinea version 5.x and standard Active Directory schema data storage model. **Note**: The difference between the Dynamic_Schema_5_0 data storage model and the CDC_RFC_2307_3 data storage model is that in the Dynamic_Schema_5_0 storage model, all Delinea data is stored as part of the zone. In the CDC_RFC_2307_3 storage model, user and group attributes are stored as part of the User and Group objects.
CDC_RFC_2307_3	9	Hierarchical RFC 2307-compatible zone, version 5.x
SFU_3_0_V5	10	Hierarchical SFU zone, version 5.x

If the zone is not in one of these formats, an exception is thrown. For more information about the difference between these different schema types and the corresponding zone types, see “Planning for data storage in Active Directory” in the *Planning and Deployment Guide*.

Exceptions

Schema throws an ApplicationException if the zone schema is not recognized.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set zone = GetZone("ajax.org/UNIX/Zones/test_lab")
'If the zone uses the SFU schema, display its domain
If zone.IsSFU = true
wScript.Echo zone.SFUDomain
end if
...
```

SFUDomain

Gets or sets the Active Directory domain associated with this zone when the zone is determined to be a zone that uses the Microsoft Services for UNIX (SFU) schema extension.

Syntax

```
string SFUDomain {get; set;}
```

Property value

The Active Directory domain for the zone.

Example

The following code sample illustrates using this method in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab,cn=Zones,cn=UNIX,dc=ajax,dc=org")
'If the zone uses the SFU schema, display its domain
If zone.IsSFU = true
wScript.Echo zone.SFUDomain
end if
...
```

UserAutoProvisioningEnabled

Indicates whether the zone has auto-provisioning of user profiles enabled.

Syntax

```
bool UserAutoProvisioningEnabled {get;}
```

Property value

Returns `true` if the zone has auto-provisioning enabled for user profiles.

Discussion

When automatic provisioning is enabled for users, the Zone Provisioning Agent can automatically provision new UNIX profiles for new Active Directory users. In addition to enabling auto-provisioning, you must specify a provisioning group to use as the source for provisioning data. For details about automatic provisioning of users and groups, see the *Planning and Deployment Guide*.

Version

Gets the version number of the data schema.

Syntax

```
int Version {get;}
```

Property value

The version number associated with the schema found.

Example

The following code sample illustrates using Version in a script:

```
...
'Specify the zone you want to work with
set zone = GetZoneByPath("LDAP://cn=test_lab/cn=Zones, cn=UNIX,dc=ajax,dc=org")

'Display the schema version number for the zone
wScript.Echo zone.Version
...
```

Delinea

HierarchicalZoneComputer

The HierarchicalZoneComputer class represents a computer joined to a hierarchical zone.

Syntax

public interface IHierarchicalZoneComputer : IComputer

Discussion

The HierarchicalZoneComputer class inherits many methods and properties from the Unexpected Link Text class, but adds support for partial profiles and inheritable roles. Under hierarchical zones, both identity (profile data) and access (authorization data) are inherited, such that a computer's effective identity or access are determined by all the profile data and all the access data at all levels of the hierarchy.

See Unexpected Link Text for a discussion of profile and access inheritance.

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager, but are referred to in the method and property descriptions where appropriate.

Methods

The HierarchicalZoneComputer class provides the following methods:

Unexpected Link Text	Adds a group to the computer.
Unexpected Link Text	Adds a computer-specific partial profile for a specified group.
Unexpected Link Text	Adds a computer-specific partial profile for a specified local group.
Unexpected Link Text	Adds a computer-specific partial profile for a specified user.
Unexpected Link Text	Adds an empty role assignment.
Unexpected Link Text	Adds a computer-specific partial profile for a specified user.
Unexpected Link Text	Commits changes to the group object to Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Creates a pending imported group in this computer.
Unexpected Link Text	Creates a pending imported user in this computer.
Unexpected Link Text	Deletes the computer profile from Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Deletes all computer-specific users and groups.
Unexpected Link Text	Deletes the computer zone object if it exists.
Unexpected Link Text	Returns a group given a role for the group.
Unexpected Link Text	Returns an enumeration of groups in the computer object.
Unexpected Link Text	Returns the Active Directory object for the computer. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Returns an enumeration of effective users under this computer zone.
Unexpected Link Text	Returns the UNIX group profile in this computer zone for the specified Active Directory group.

Unexpected Link Text	Returns the UNIX group profile in this computer zone for the Active Directory group specified by distinguished name.
Unexpected Link Text	Returns the UNIX group profile in this computer zone for the Active Directory group specified by group name.
Unexpected Link Text	Returns an enumeration of the UNIX groups in this computer zone.
Unexpected Link Text	Returns the group with the specified ID pending import.
Unexpected Link Text	Returns an enumeration of groups pending import to this computer zone.
Unexpected Link Text	Returns the user with the specified ID pending import.
Unexpected Link Text	Returns an enumeration of users pending import to this computer zone.
Unexpected Link Text	Returns the numeric identifier for the pending import group with the specified group name.

| method | Description | | Unexpected Link Text | Returns the local UNIX group profile for a specified group name in the zone. | | Unexpected Link Text | Returns a local group profile using the distinguished name (DN) of the profile. | | Unexpected Link Text | Returns the local group profile using the Group Identifier (GID). This method is exposed to the .COM interface. | | Unexpected Link Text | Returns a list of the local group profiles in the zone. | | Unexpected Link Text | Returns the local user profile using the specified user name. | | Unexpected Link Text | Returns the local user profile specified by the distinguished name (DN) of the profile. | | Unexpected Link Text | Returns the local user profile using the User Identifier (UID). This method is exposed to the .COM interface | | Unexpected Link Text | Returns a list of the local user profiles in the zone. | | Unexpected Link Text | Returns the numeric identifier for the pending import user with the specified user name. | | Unexpected Link Text | VBScript interface to access NSS variables. | | Unexpected Link Text | VBScript interface to obtain all NSS variable names. | | Unexpected Link Text | Returns the primary profile for the specified user. | | Unexpected Link Text | Returns the role assignment for the specified role and trustee. | | Unexpected Link Text | Returns the role assignment for the specified GUID. | | Unexpected Link Text | Returns the collection of role assignments in the computer. | | Unexpected Link Text | Returns the role assignment given to all Active Directory users who have a specified role. | | Unexpected Link Text | Returns the role assignment given to all UNIX users who have a specified role. | | Unexpected Link Text | Returns an enumeration of the secondary profiles for the specified user. | | Unexpected Link Text | Returns an enumeration of all the user profiles for the specified user. | | Unexpected Link Text | Returns an enumeration of all the user role assignments in this computer zone. | | Unexpected Link Text | Returns the UNIX user profile in this computer zone for the specified user. | | Unexpected Link Text | Returns the UNIX user profile in this computer zone for the user specified by distinguished name. | | Unexpected Link Text | Returns the UNIX user profile in this computer zone for the user specified by user name. | | Unexpected Link Text | Returns the UNIX user profile in this computer zone for the user specified by UID. | | Unexpected Link Text | Indicates whether the specified user has a profile in this computer zone. |

Properties

The HierarchicalZoneComputer class provides the following properties:

Unexpected Link Text	Gets the IADs interface of the zone object in Active Directory. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the LDAP path to the zone object. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the Active Directory client version number. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the canonical name of the computer object. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the LDAP path of the computer zone object.

Unexpected Link Text	Indicates whether the CIMs data associated with this object is orphaned by the current credentials. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether this computer is an orphan zone object.
Unexpected Link Text	Indicates whether the CIMS data associated with this object is readable with the current user credentials. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Indicates whether the CIMS data associated with this object is writable with the current user credentials. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the computer is enabled for JBoss. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the name of the computer object. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the map of profile variables.
Unexpected Link Text	Gets the LDAP path to the computer UNIX profile. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets the version of the data schema. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the computer is enabled for Tomcat. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the UNIX directory path that is used to substitute for % in user profiles.
Unexpected Link Text	Gets or sets the shell that is used to substitute for %{shell}in user profiles.
Unexpected Link Text	Gets the version number of the data schema. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the computer is enabled for WebLogic. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Determines whether the computer is enabled for WebSphere. (Inherited from Unexpected Link Text.)
Unexpected Link Text	Gets or sets the zone that this computer joins.
Unexpected Link Text	Gets the zone mode of the computer. (Inherited from Unexpected Link Text.)

AddAccessGroup

Adds a group to the computer.

Syntax

IMzRoleAssignment AddAccessGroup(DirectoryEntry groupDE)

IMzRoleAssignment AddAccessGroup(SearchResult groupSR)

IMzRoleAssignment AddAccessGroup(string groupDn)

IMzRoleAssignment AddAccessGroup(IAdsGroup groupIAds)

Parameters

Specify one of the following parameters when using this method.

groupDE	The directory entry for the group you want to add.
groupSr	The directory entry for a group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The computer role assignment that includes the specified group (IMzRoleAssignment.TrusteeType==Group).

Discussion

The role assignment is not stored in Active Directory until you call the Unexpected Link Text method.

The AddAccessGroup(DirectoryEntry groupDE) and AddAccessGroup(SearchResult groupSr) methods are available only for .NET-based programs. Call Unexpected Link Text for VBScript.

Exceptions

AddAccessGroup may throw one of the following exceptions:

- ApplicationException if the specified parameter is not a group or the method cannot find the group.
- ArgumentNullException if you pass a null parameter.

AddGroupPartialProfile

Adds a computer-specific partial profile for the specified group to the computer.

Syntax

```
IHierarchicalGroup AddGroupPartialProfile(DirectoryEntry groupDE)
IHierarchicalGroup AddGroupPartialProfile(SearchResult groupSR)
IHierarchicalGroup AddGroupPartialProfile(string groupDn)
IHierarchicalGroup AddGroupPartialProfile(IAdsGroup groupIAds)
```

Parameters

Specify one of the following parameters when using this method.

groupDE	The directory entry for the group for which you want a partial profile.
groupSr	The directory entry for a group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The hierarchical group object that represents the group profile.

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager.

This method creates a computer zone and a new group profile with values set for the Cims and User properties. You can then add other properties to the profile.

The profile is not stored in Active Directory until you call the Unexpected Link Text method.

The AddGroupPartialProfile(DirectoryEntry groupDE) and AddGroupPartialProfile(SearchResult groupSr) methods are available only for .NET-based programs.

Exceptions

If you pass a null or empty parameter, AddGroupPartialProfile throws the exception ArgumentNullException.

Example

The HierarchicalZoneComputer.AddGroupPartialProfile method is used in the same way as the HierarchicalZone.AddGroupPartialProfile method. See Unexpected Link Text for an example.

AddLocalGroupPartialProfile

Adds a partial profile for the specified group to the zone.

Syntax

```
IHierarchicalUser AddlocalGroupPartialProfile(string groupName)
```

Parameters

Specify groupName; the name of the local group.

Return value

The hierarchical group object that represents the local group profile.

Exceptions

If you pass a null parameter, AddLocalGroupPartialProfile throws the exception ArgumentNullException.

AddLocalUserPartialProfile

Adds a partial profile for the specified user to the zone.

Syntax

```
IHierarchicalUser AddLocalUserPartialProfile(string userName)
```

Parameters

Specify `userName`; the user name of the local user.

Return value

The hierarchical user object that represents the local user profile.

Exceptions

If you pass a `null` parameter, `AddLocalUserPartialProfile` throws the exception `ArgumentNullException`.

AddRoleAssignment

Adds an empty role assignment to the computer.

Syntax

```
IRoleAssignment AddRoleAssignment()
```

Return value

An empty role assignment object. This role assignment is not stored in Active Directory until you call the RoleAssignment:[Commit](dev/windows-api/object-reference/computerrole/commit.md) method.

AddUserPartialProfile

Adds a computer-specific partial profile for the specified user to the computer.

Syntax

```
IHierarchicalUser AddUserPartialProfile(DirectoryEntry userDE)
IHierarchicalUser AddUserPartialProfile(SearchResult userSR)
IHierarchicalUser AddUserPartialProfile(string userDn)
IHierarchicalUser AddUserPartialProfile(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want a partial profile.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The hierarchical user object that represents the user profile.

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager.

This method creates a computer zone and a new user profile with values set for the `Cims` and `User` properties. You can then add other properties to the profile.

The profile is not stored in Active Directory until you call the Unexpected Link Text method.

The `AddUserPartialProfile(DirectoryEntry userDE)` and `AddUserPartialProfile(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

If you pass a `null` or empty parameter, `AddUserPartialProfile` throws the exception `ArgumentNullException`.

Example

The `HierarchicalZoneComputer.AddUserPartialProfile` method is used in the same way as the `HierarchicalZone.AddUserPartialProfile` method. See Unexpected Link Text for an example.

CreateImportPendingGroup

Creates a "pending import" group in this computer.

Syntax

```
IGroupInfo CreateImportPendingGroup (string source, DateTime timestamp)
```

Parameters

Specify the following parameters when using this method.

source	The location of the source data for the group to be imported.
timestamp	The date and time at which the data was retrieved.

Return value

The newly created pending import group object.

Discussion

Group profiles in a pending import group object needed to be mapped to Active Directory groups before they can be used. Groups in this state are normally imported from NIS domains or from text files and stored temporarily either in Active Directory or XML files until they are mapped to Active Directory accounts. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

CreateImportPendingUser

Creates a "pending import" user in this computer.

Syntax

```
IUserInfo CreateImportPendingUser(string source, DateTime timestamp)
```

Parameters

Specify the following parameters when using this method.

source	The location of the source data for the user to be imported.
timestamp	The date and time at which the data was retrieved.

Return value

The newly-created pending import user object.

Discussion

User profiles in a pending import user object need to be mapped to Active Directory groups before they can be used. Users in this state are normally imported from NIS domains or from text files and stored temporarily either in Active Directory or in XML files until they are mapped to Active Directory accounts. For more information about importing and mapping users, see the *Administrator's Guide for Linux and UNIX*.

DeleteAllProfiles

Deletes all computer-specific users and groups.

Syntax

```
void DeleteAllProfiles ()
```

Discussion

Deleting a computer (Unexpected Link Text) doesn't delete all the profiles associated with the computer. Use this method to delete computer-specific profiles after deleting the computer.

DeleteZone

Deletes the computer zone object.

Syntax

```
void DeleteZone ()
```

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager.

This method deletes only the computer zone object (if it exists). Call the Unexpected Link Text method to delete the computer profile object.

GetAccessGroup

Gets a user group assigned to this computer given a specific role.

Syntax

```
IMzRoleAssignment GetAccessGroup(IRole role, DirectoryEntry group)
IMzRoleAssignment GetAccessGroup(IRole role, SearchResult groupSr)
IMzRoleAssignment GetAccessGroup(IRole role, string groupDn)
IMzRoleAssignment GetAccessGroup(IRole role, IADsGroup groupIAds)
```

Parameters

Specify the following parameter when using this method:

role	The role of the group.

Specify one of the following parameters when using this method.

group	The directory entry for the group.
groupSr	The directory entry for the group specified as a search result.
groupDn	The group specified as a distinguished name.
groupIads	The IADs interface to the group.

Return value

The computer role assignment that includes the specified group (IMzRoleAssignment.TrusteeType==Group).

Discussion

Any number of user groups can be assigned to a computer role and each of those groups can have more than one role. Use this method to get the computer role assignment for a specific group and role.

The GetAccessGroup(IRole role, DirectoryEntry groupDE) and GetAccessGroup(IRole role, SearchResult groupSr) methods are available only for .NET-based programs. Call Unexpected Link Text for VBScript.

Exceptions

GetAccessGroup may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null.
- ApplicationException if the parameter value is not a valid user; or if it failed to create a role assignment because it cannot find the user.

Example

The HierarchicalZoneComputer.GetAccessGroup method is used in the same way as the HierarchicalZone.GetAccessGroup method. See Unexpected Link Text for an example of using the HierarchicalZone.GetAccessGroup method in a script.

GetAccessGroups

Returns the computer roles assigned to this computer.

Syntax

```
IRoleAssignments GetAccessGroups()
```

Return value

The collection of computer roles. Enumerate this object to get all of the `IAzRoleAssignment` objects for this computer.

GetEffectiveUserUnixProfiles

Returns the collection of effective users under this computer zone.

Syntax

```
IUserUnixProfiles GetEffectiveUserUnixProfiles()
```

Return value

A collection of IHierarchicalUser objects representing all the user profiles under this computer zone, including those inherited from zones higher in the hierarchy.

GetGroupUnixProfile

Returns the hierarchical group profile for a specified Active Directory group in the computer zone.

Syntax

```
IHierarchicalGroup GetGroupUnixProfile(IGroup group)
```

Parameter

Specify the following parameter when using this method:

group	The group for which you want profile information.

Return value

The profile for the specified group, or `null` if none is found.

Discussion

This method uses the `Centrify.DirectControl.API.IGroup` group returned by a Unexpected Link Text or Unexpected Link Text call to retrieve the group profile.

Exceptions

`GetGroupUnixProfile` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.
- `ApplicationException` if there is more than one group with the specified `IGroup` value in the zone.

GetGroupUnixProfileByDN

Returns the UNIX profile for a group in this computer zone using the distinguished name (DN) of the profile.

Syntax

```
IHierarchicalGroup GetGroupUnixProfileByDN(string dn)
```

Parameter

Specify the following parameter when using this method:

dn	The distinguished name (DN) of the group profile.

Return value

The group profile with the distinguished name (DN) specified, or `null` if no matching group profile is found.

Discussion

The group profile is the service connection point associated with the Active Directory group object.

Exceptions

`GetGroupUnixProfileByDN` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.

GetGroupUnixProfileByName

Returns the hierarchical group profile for a group with the specified name in the computer zone.

Syntax

```
IHierarchicalGroup GetGroupUnixProfileByName(string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the UNIX group profile for which you want to retrieve information.

Return value

The profile for the specified group name, or `null` if no profile is found.

Discussion

The name you specify should be the UNIX group name for the group if it differs from the Active Directory name for the group.

Exceptions

`GetGroupUnixProfileByName` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null` or empty.
- `NotSupportedException` if the computer zone schema is not supported.
- `ApplicationException` if there is more than one group with the specified name in the zone.

GetGroupUnixProfiles

Returns the collection of UNIX group profiles that have been defined for the computer zone.

Syntax

```
IComputerGroupUnixProfiles GetGroupUnixProfiles()
```

Return value

Returns a collection of Unexpected Link Text objects.

GetImportPendingGroup

Returns a group pending import to the computer zone given the GUID.

Syntax

```
IGroupInfo GetImportPendingGroup(string id, bool storePendingAD, string storePendingFilePath)
```

Parameter

Specify the following parameters when using this method:

id	The GUID of the group that's pending import.
storePendingAD	Specify true if the group is being imported from Active Directory. Specify false if the group is being imported from an XML file.
storePendingFilePath	The file path of the XML file.

Return value

The IGroupInfo object for the specified group.

Discussion

This method takes the storePendingAD Boolean and the storePendingFilePath information, stores them in the group profile, then finds and returns the group pending import that has the specified GUID.

Group profiles that are pending import are normally imported from NIS domains or from text files and not yet mapped to Active Directory groups. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

GetImportPendingGroups

Returns the list of groups pending import to this computer zone.

Syntax

```
IGroupInfos GetImportPendingGroups()
```

Return value

The collection of group profiles pending import to this computer zone.

GetImportPendingUser

Returns an individual user pending import for this computer given the GUID.

Syntax

```
IUserInfo GetImportPendingUser(string id, bool storePendingAD, string storePendingFilePath)
```

Parameter

Specify the following parameters when using this method:

id	The GUID of the user that's pending import.
storePendingAD	Specify true if the user is being imported from Active Directory. Specify false if the user is being imported from an XML file.
storePendingFilePath	The file path of the XML file.

Return value

The IUserInfo object for the specified ID in the computer.

Discussion

This method takes the storePendingAD Boolean and the storePendingFilePath information, stores them in the user profile, then finds and returns the user pending import that has the specified GUID.

User profiles that are pending import are normally imported from NIS domains or from text files and not yet mapped to Active Directory groups. For more information about importing and mapping groups, see the *Administrator's Guide for Linux and UNIX*.

GetImportPendingUsers

Returns the collection of users pending import to this computer zone.

Syntax

```
IUserInfos GetImportPendingUsers()
```

Return value

The collection of user profiles pending import to this computer zone.

GetIPendingGroupID

Returns the numeric identifier of the pending import group.

Syntax

```
IGroupInfo GetPendingGroupID()
```

Return value

The pending import group specified by group name to the selected zone.

GetIPendingUserID

Returns the numeric identifier of the pending import user.

Syntax

```
IUserInfo GetPendingUserID()
```

Return value

The pending import user specified by the user name to the selected zone.

GetLocalGroupUnixProfile

Returns the UNIX group profile for a specified local group.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfile(string groupName)
```

Parameter

Specify the following parameter when using this method:

`groupName	The name of the local group for which you want to retrieve profile information.

Return value

The Unexpected Link Text object for the specified local group name. If there is no group, `null` is returned.

Exceptions

`GetGroupUnixProfile` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.

GetLocalGroupUnixProfileByDN

Returns the Local UNIX profile for a group in the zone using the distinguished name (DN) of the profile.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfileByDN(string dn)
```

Parameter

Specify the following parameters when using this method.

dn	The distinguished name (DN) of the local group profile.

Return value

The local group profile with the distinguished name (DN) matching the distinguished name specified, or `null` if no matching group profile is found.

GetLocalGroupUnixProfileByGid (Int32)

Returns the Local UNIX profile for a group in the zone using the group identifier (GID) of the profile. This method is exposed to the .COM interface.

Syntax

```
IGroupUnixProfile GetLocalGroupUnixProfileByGid(int gid)
```

Parameter

Specify the following parameters when using this method.

gid	The group identifier (GID) of the local group profile.

Return value

The local group profile with the specified group identifier (GID) or `null` if no matching group profile is found.

GetLocalGroupUnixProfiles

Get the list of local group profiles in the zone.

Syntax

```
IGroupUnixProfiles GetLocalGroupUnixProfiles()
```

Return value

Returns a collection of Unexpected Link Text objects. If there are no groups, `null` is returned.

GetLocalUserUnixProfile

Returns the UNIX user profile for a specified local group.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfile(string userName)
```

Parameter

Specify the `userName` parameter when using this method.

Return value

Returns the local user profile with the specified user name. If there is no group, `null` is returned.

GetLocalUserUnixProfileByDN

Returns the local UNIX profile for a user in the zone using the distinguished name (DN) of the profile.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfileByDN(string dn)
```

Parameter

Specify the following parameters when using this method.

dn	The distinguished name (DN) of the local user profile.

GetLocalUserUnixProfileByUid (Int32)

Returns the local UNIX profile for a user in the zone using the user identifier (GID) of the profile. This method is exposed to the .COM interface.

Syntax

```
IUserUnixProfile GetLocalUserUnixProfileByUid(int uid)
```

Parameter

Specify the following parameters when using this method.

uid	The user identifier (UID) of the local user profile.

Return value

The local user profile with the specified user identifier (UID) or `null` if no matching user profile is found.

GetLocalUserUnixProfiles

Get a list of local UNIX user profiles in the zone.

Syntax

```
IUserUnixProfiles GetLocalUserUnixProfiles()
```

Return value

Returns a collection of local user profiles in the zone. If there are no users, `null` is returned.

GetNssVariable

Returns the specified NSS environment variable; VBScript only.

Syntax

```
string GetNssVariable (string name)
```

Parameter

Specify the following parameter when using this method:

name	The name of the variable.

Return value

The value of the variable, or `null` if there is no NSS variable with the specified name.

GetNSSVariables

Returns the names of all NSS variables; VBScript only.

Syntax

```
IEnumerable GetNSSVariables()
```

Return value

Returns a collection of NSS variable names.

GetPrimaryUser

Returns the primary profile for the specified user.

Syntax

```
IHierarchicalUser GetPrimaryUser(DirectoryEntry userDE)
IHierarchicalUser GetPrimaryUser(SearchResult userSR)
IHierarchicalUser GetPrimaryUser(string userDn)
IHierarchicalUser GetPrimaryUser(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the primary profile.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The hierarchical user object that represents the user profile.

Discussion

The primary profile is the profile at the highest level in the zone hierarchy where the user's profile is defined. All or part of the primary profile can be overridden by secondary profiles farther down in the hierarchy.

The GetPrimaryUser(DirectoryEntry userDE)andGetPrimaryUser(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

`GetPrimaryUser` throws an `ArgumentNullException` if the specified parameter value is `null` or empty.

GetRoleAssignment

Returns a role assignment given a role and trustee.

Syntax

```
IRoleAssignment GetRoleAssignment (IRole role, string dn)
```

Parameter

Specify the following parameters when using this method.

role	The role for which you want the assignment.
dn	The distinguished name of the user or group to whom the role is assigned.

Return value

The role assignment, or `null` if no match is found.

Exceptions

`GetRoleAssignment` throws an `ArgumentNullException` if either specified parameter value is `null` or empty.

GetRoleAssignmentById

Returns a role assignment given an ID.

Syntax

```
IRoleAssignment GetRoleAssignment (Guid id)
```

Parameter

Specify the following parameter when using this method:

id	The GUID of the role assignment.

Return value

The role assignment, or `null` if no match is found.

Exceptions

`GetRoleAssignmentById` throws an `ArgumentNullException` if the specified parameter value is empty.

GetRoleAssigments

Returns all the role assignments in the computer.

Syntax

```
IRoleAssignments GetRoleAssignments()
```

Return value

The collection of role assignments in the computer.

GetRoleAssignmentToAllADUsers

Returns the role assignment given to all Active Directory users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToAllADUsers(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignment.

Return value

The role assignment for the specified role.

Exceptions

GetRoleAssignmentToAllADUsers throws an ArgumentNullException if the specified parameter value is null.

GetRoleAssignmentToAllUnixUsers

Returns the role assignment given to all UNIX users who have a specified role.

Syntax

```
IRoleAssignment GetRoleAssignmentToAllUnixUsers(IRole role)
```

Parameter

Specify the following parameter when using this method:

role	The user role for which you want the role assignment.

Return value

The role assignment for the specified role.

Discussion

This method returns the role assignment for all local UNIX users with the specified role.

Exceptions

GetRoleAssignmentToAllUnixUsers throws an ArgumentNullException if the specified parameter value is null.

GetSecondaryUsers

Returns the secondary profiles for the specified user.

Syntax

```
IUserUnixProfiles GetSecondaryUsers(DirectoryEntry userDE)
IUserUnixProfiles GetSecondaryUsers(SearchResult userSR)
IUserUnixProfiles GetSecondaryUsers(string userDn)
IUserUnixProfiles GetSecondaryUsers(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the secondary profiles.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The collection of secondary user UNIX profiles.

Discussion

The primary profile is the profile at the highest level in the zone hierarchy where the user's profile is defined. All or part of the primary profile can be overridden by secondary profiles farther down in the hierarchy.

The `GetSecondaryUsers(DirectoryEntry userDE)` and `GetSecondaryUsers(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

`GetSecondaryUsers` throws an `ArgumentNullException` if the specified parameter value is `null` or the user does not exist.

GetUserProfiles

Returns all the profiles for the specified user.

Syntax

```
IUserUnixProfiles GetUserProfiles(DirectoryEntry userDE)
IUserUnixProfiles GetUserProfiles(SearchResult userSR)
IUserUnixProfiles GetUserProfiles(string userDn)
IUserUnixProfiles GetUserProfiles(IAdsUser userIAds)
```

Parameters

Specify one of the following parameters when using this method.

userDE	The directory entry for the user for which you want the profiles.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.

Return value

The collection of user UNIX profiles.

Discussion

The GetUserProfiles(DirectoryEntry userDE) and GetUserProfiles(SearchResult userSr) methods are available only for .NET-based programs.

Exceptions

GetUserProfiles throws an ArgumentNullException if the specified parameter value is null or the user does not exist.

GetUserRoleAssignments

Returns all the user role assignments in the computer zone, or for a specific user in the computer zone.

Syntax

```
IRoleAssignments GetUserRoleAssignments()
IRoleAssignments GetUserRoleAssignments(DirectoryEntry userDE)
IRoleAssignments GetUserRoleAssignments(SearchResult userSR)
IRoleAssignments GetUserRoleAssignments(string userDn)
IRoleAssignments GetUserRoleAssignments(IAdsUser userIAds)
IRoleAssignments GetUserRoleAssignments(IUser user)
```

Parameters

Specify no parameters to return all the role assignments in the computer zone.

Specify one of the following parameters to return all the role assignments for a specific user:

userDE	The directory entry for the user for which you want the role assignments.
userSr	The directory entry for a user specified as a search result.
userDn	The user specified as a distinguished name.
userIads	The IADs interface to the user.
user	The user specified as a CIMS user object.

Return value

The collection of role assignments as `IRoleAssignment` objects.

Discussion

The `GetUserRoleAssignments(DirectoryEntry userDE)` and `GetUserRoleAssignments(SearchResult userSr)` methods are available only for .NET-based programs.

Exceptions

`GetUserRoleAssignments` throws an `ArgumentNullException` if the required parameter is `null` or empty.

GetUserUnixProfile

Returns the UNIX user profile for a specified Active Directory user in this computer zone.

Syntax

```
IHierarchicalUser GetUserUnixProfile(IUser user)
```

Parameter

Specify the following parameter when using this method:

user	The user object for Active Directory user.

Return value

The profile for the specified Active Directory user, or `null` if none could be found.

Discussion

This method uses the `Centrify.DirectControl.API.IUser` returned by a `GetUser` or `GetUserByPath` call to retrieve the user profile.

Exceptions

`GetUserUnixProfile` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.

GetUserUnixProfileByDN

Returns the UNIX profile for a user in this computer zone given the distinguished name (DN) of the profile.

Syntax

```
IHierarchicalUser GetUserUnixProfileByDN(string dn)
```

Parameter

Specify the following parameter when using this method:

dn	The distinguished name (DN) of the user profile.

Return value

The user profile with the distinguished name (DN) specified, or `null` if no matching user profile is found.

Discussion

The user profile is the service connection point associated with the Active Directory user object.

Exceptions

`GetUserUnixProfileByDN` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.

GetUserUnixProfileByName

Returns the UNIX profile for a user in this computer zone given the user name.

Syntax

```
IHierarchicalUser GetUserUnixProfileByName(string name)
```

Parameter

Specify the following parameter when using this method:

name	The user's UNIX login name.

Return value

The profile of the specified user, or null if none is found.

Exceptions

GetUserUnixProfileByName may throw one of the following exceptions:

- ArgumentNullException if the specified parameter value is null or empty.
- NotSupportedException if the computer zone schema is not supported.
- ApplicationException if there is more than one user with the specified name in the zone.

GetUserUnixProfileByUid

Returns the UNIX profile for a user in this computer zone given the user identifier (UID).

Syntax

```
IHierarchicalUser GetUserUnixProfileByUid(int uid)
IHierarchicalUser GetUserUnixProfileByUid(long uid)
```

Parameter

Specify the following parameter when using this method:

uid	The user identifier (UID) associated with the Active Directory user.

Return value

The user profile for the specified UID in the computer zone, or `null` if none is found.

Discussion

If there are multiple user profiles with the UID specified, this method returns only the first user profile found. To find all user profiles with a specific UID, use the Unexpected Link Text method to return the collection of profiles for a computer, then search the collection for the UID.

> **Note**: There are two versions of this method: one designed for COM-based programs that supports a 32-bit signed number for the uid argument and one designed for .NET-based programs that allows a 64-bit signed number for the uid argument. These methods are provided for backward compatibility with earlier versions of Delinea software. These methods are not applicable for version 4.0 or later.

Exceptions

`GetUserUnixProfileByUid` may throw one of the following exceptions:

- `ArgumentException` if you specify a negative UID.
- `NotSupportedException` if the computer zone schema is not supported.

GetUserUnixProfiles

Returns the list of UNIX user profiles in this computer zone.

Syntax

```
IComputerUserUnixProfiles GetUserUnixProfiles()
```

Return value

The collection of computer user UNIX profiles.

GroupUnixProfileExists

Indicates whether a UNIX profile exists for the specified group in this computer zone.

Syntax

```
bool GroupUnixProfileExists(IGroup group)
```

Parameter

Specify the following parameter when using this method:

group	The group for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if a UNIX profile is found in this computer zone for the specified group, or `false` if no UNIX profile exists for the group in the computer zone.

Exceptions

`GetUserUnixProfileByUid` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.

LocalGroupUnixProfileExists

Checks whether a UNIX profile exists for the specified local group in the zone.

Syntax

```
bool LocalGroupUnixProfileExists(string groupName)
```

Parameter

Specify the following parameter when using this method:

groupName	The group name for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if the local UNIX group profile is found in the zone, or `false` if no UNIX profile exists for the group in the zone.

Exceptions

`LocalGroupUnixProfileExists` may throw the following exception:

- `ArgumentNullException` if the specified parameter value is `null`.

LocalUserUnixProfileExists

Checks whether a local UNIX profile exists for the specified user in the zone.

Syntax

```
bool LocalUserUnixProfileExists(string userName)
```

Parameter

Specify the following parameter when using this method:

userName	The local user name for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if a UNIX profile is found in the zone for the specified local user, or `false` if no UNIX profile exists for the user in the zone.

Exceptions

`UserUnixProfileExists` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.

SetNSSVariable

Sets the value of the specified NSS environment variable; VBScript only.

Syntax

```
string SetNssVariable (string name, string value)
```

Parameter

Specify the following parameters when using this method.

name	The name of the variable.
value	The value of the variable. Pass null to remove the variable.

UserUnixProfileExists

Indicates whether a UNIX profile exists for the specified user in this computer zone.

Syntax

```
bool UserUnixProfileExists(IUser user)
```

Parameter

Specify the following parameter when using this method:

user	The user name for which you want to check whether a UNIX profile exists.

Return value

Returns `true` if a UNIX profile is found for the specified user, or `false` if no UNIX profile exists for the user in the computer zone.

Exceptions

`UserUnixProfileExists` may throw one of the following exceptions:

- `ArgumentNullException` if the specified parameter value is `null`.
- `NotSupportedException` if the computer zone schema is not supported.

ComputerZoneADsPath

Gets the LDAP path of the computer zone object.

Syntax

```
string ComputerZoneADsPath {get;}
```

Property value

The LDAP path.

Discussion

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager.

If the computer zone does not contain any users, groups, or computer roles, this property returns `null` or `string.Empty`.

IsOrphanZone

Indicates whether this computer zone is an orphan zone.

Syntax

```
bool OrphanZone {get;}
```

Property value

Returns `true` if this computer zone is an orphan zone object.

Discussion

The computer zone is an orphan if it has no corresponding service connection point (SCP) and Active Directory computer object.

When you assign computer-level overrides for user, group, or computer role assignments, Delinea creates a *computer zone*, which is a special type of zone that contains the users, groups, and computer role assignments that are specific to only that one computer. Computer zones are not exposed as zones in Access Manager.

NssVariables

Gets all the NSS environment variables.

Syntax

```
IDictionary\<string, string\> NssVariables {get;}
```

Property value

A dictionary of key-value pairs that define all the profile variables.

Discussion

This property uses a 64-bit value for use in .NET modules. Use the GetNSSVariablesproperty for VBScript.

UserHomeDirectory

Gets or sets the UNIX directory path that is used to substitute for % in user profiles.

Syntax

```
string UserHomeDirectory {get; set;}
```

Property value

The UNIX directory path, or `null` to inherit the path from the parent zone.

UserShell

Gets or sets the shell that is used to substitute for % in user profiles.

Syntax

```
int UserDefaultGid {get; set;}
```

Property value

The user shell, or `null` to inherit the shell from the parent zone.

Zone

Determines the zone object for the zone to which the computer is currently joined.

Syntax

```
IHierarchicalZone Zone {get; set;}
```

Property value

The zone the computer account has joined.

Discussion

Each computer object can only be associated with one zone: the zone used to join the computer to its Active Directory domain.

Exceptions

Zone throws an `ApplicationException` if you try to set the zone to a type that is not hierarchical or is not supported or if the computer is already joined to a zone and you try set a new zone.

Timebox

A role specifies a collection of rights. A role object contains a field, timebox, that defines the hours and days of the week that a role is either enabled or disabled. Setting the timebox field in a role object defines when a role's rights are in effect.

You can read and set a role's timebox field using the Unexpected Link Text property. You can modify an existing timebox value one day or one hour at a time using the Unexpected Link Text and Unexpected Link Text methods.

To interpret a timebox value, or to set it directly, however, you must know the timebox value format. This appendix explains following formats:

- Hex string
- Hour mapping
- Day mapping

Hex string

The timebox value is a 42-character (21-byte) hexadecimal value stored as a string. When the hex value is converted to a binary value, its 168 bits each map to a single hour within the week. If a bit is set to 1, its corresponding hour is enabled for the role. If set to 0, its corresponding hour is disabled.

Hour mapping

Each day of the week takes three bytes (24 bits) to specify how its hours are enabled or disabled. The following tables show how the hours of a day are mapped to the bits within each of a day's three bytes.

For byte 0 of each day of the week, you can enable or disable the hours a role is available from midnight to 8:00 AM:

12-1 AM	0 (least-significant bit)
1-2 AM	1
2-3 AM	2
3-4 AM	3
4-5 AM	4
5-6 AM	5
6-7 AM	6
7-8 AM	7 (most-significant bit)

For byte 1 of each day of the week, you can enable or disable the hours a role is available from 8:00 AM to 4:00 PM:

8-9 AM	0 (least-significant bit)
9-10 AM	1
10-11 AM	2
11-12 AM	3
12-1 PM	4
1-2 PM	5

2-3 PM	6
3-4 PM	7 (most-significant bit)

For byte 2 of each day of the week, you can enable or disable the hours a role is available from 4:00 PM to midnight:

4-5 PM	0 (least-significant bit)
5-6 PM	1
6-7 PM	2
7-8 PM	3
8-9 PM	4
9-10 PM	5
10-11 PM	6
11-12 PM	7 (most-significant bit)

Day mapping

Each of the seven days in a week have three bytes within the 21-byte timebox value. These bytes are in chronological order from most-significant byte to least-significant byte. (Note that this is the opposite of chronological bit order within each byte, which is LSB to MSB.)

The starting point of a week is 4 PM on Saturday afternoon.

The table below shows how each day's three bytes (0-2) map to the timebox value's bytes, listed here in order from most-significant byte to least-significant byte.

Saturday, byte 2	20 (most-significant byte)
Sunday, byte 0	19
Sunday, byte 1	18
Sunday, byte 2	17
Monday, byte 0	16
Monday, byte 1	15
Monday, byte 2	14
Tuesday, byte 0	13
Tuesday, byte 1	12

Tuesday, byte 2	11
Wednesday, byte 0	10
Wednesday, byte 1	9
Wednesday, byte 2	8
Thursday, byte 0	7
Thursday, byte 1	6
Thursday, byte 2	5
Friday, byte 0	4
Friday, byte 1	3
Friday, byte 2	2
Saturday, byte 0	1
Saturday, byte 1	0 (least-significant byte)

Server Suite - All Release Notes

2022.1 Release

- Server Suite Release Notes
- Server Suite Agent Release Notes
- Server Suite Authentication and Privilege Elevation Release Notes
- Server Suite Auditing Release Notes
- Server Suite PuTTY Release Notes
- Server Suite Mac Release Notes

For a list of supported components for the current release, see Component Versions.

2022 Release

- Server Suite Release Notes
- Server Suite Agent Release Notes
- Server Suite Authentication and Privilege Elevation Release Notes
- Server Suite Auditing Release Notes
- Server Suite PuTTY Release Notes
- Server Suite Adbindproxy Release Notes
- Server Suite Mac Release Notes

You can find release notes and documentation related to previous releases at Previous Releases.

Delinea

Server Suite Product Component Version Table

2022.1	August 2022	5.9.1	5.9.1	5.9.1	5.9.1	5.9.1	5.9.1	*	5.9.1
2022	April 2022	5.9.0	5.9.0	5.9.0	5.9.0	5.9.0	5.9.0	*	5.9.0
2021.1	December 2021	5.8.1	5.8.1	5.8.1	5.8.1	5.8.1	5.8.1	*	5.8.1
2021	July 2021	5.8.0	5.8.0	5.8.0	5.8.0	5.8.0	5.8.0	*	5.8.0
2020.1	December 2020	5.7.1	5.7.1	5.7.1	3.7.1	3.7.1	3.7.1	*	5.7.0
2020	September 2020	5.7.0	5.7.0	5.7.0	3.7.0	3.7.0	3.7.0	*	5.7.0
19.9	December 2019	5.6.1	5.6.1	5.6.1	3.6.1	3.6.1	3.6.1	*	5.6.1
19.6	August	5.6.0	5.6.0	5.6.0	3.6.0	3.6.0	3.6.0	*	5.6.0
19.2	February 2019	5.5.2	5.5.3	5.5.2	3.5.2	3.5.2	3.5.2	*	5.5.2
18.11	December 2018	5.5.2	5.5.2	5.5.2	3.5.2	3.5.2	3.5.2	5.5.2	5.5.2
18.8	August 2018	5.5.1	5.5.1	5.5.1	3.5.1	3.5.1	3.5.1	5.5.1	5.5.1
2018	April 2018	5.5.0	5.5.0	5.5.0	3.5.0	3.5.0	3.5.0	5.5.0	5.5.0
2017.3	December 2017	5.4.3	5.4.3	5.4.3	3.4.3	3.4.3	3.4.3	5.4.3	5.4.3
2017.2	September 2017	5.4.2	5.4.2	5.4.2	3.4.2	3.4.2	3.4.2	5.4.2	5.4.2
2017.1	May 2017	5.4.1	5.4.1	5.4.1	3.4.1	3.4.1	3.4.1	5.4.1	5.1.10
2017	February 2017	5.4.0	5.4.0	5.4.0	3.4.0	3.4.0	3.4.0	5.4.0	5.1.9
2016.1 Update	August 2016	5.3.1	5.3.1 Update	5.3.1	3.3.1	3.3.1	3.3.1	5.3.1	5.1.8
2016.1	April 2016	5.3.1	5.3.1	5.3.1	3.3.1	3.3.1	3.3.1	5.3.1	5.1.8
2016 Update	March 2016	5.3.0	5.3.0 Update	5.3.0	3.3.0	3.3.0 Update	3.3.0	5.3.0	5.1.7
2016	December 2015	5.3.0	5.3.0	5.3.0	3.3.0	3.3.0	3.3.0	5.3.0	5.1.7
2015.1 Update	March 2016	5.2.3	5.2.3 Update	5.2.3	3.2.3	3.2.3 Update	3.2.3	5.2.3	5.1.6
2015.1	July 2015	5.2.3	5.2.3	5.2.3	3.2.3	3.2.3	3.2.3	5.2.3	5.1.6
2015	February 2015	5.2.2	5.2.2	5.2.2	3.2.2	3.2.2	3.2.2	5.2.2	5.1.5
2014.1 Update	November 2014	5.2.0	5.2.1	5.1.4	3.2.1	3.2.1	3.2.1	5.2.0	5.1.4

2014.1	August 2014	5.2.0	5.2.0	5.1.4	3.2.1	3.2.1	3.2.1	5.2.0	5.1.4
2014	February 2014	5.1.3	5.1.3	5.1.3	3.2.0	3.2.0	3.2.0	5.1.3	5.1.3

The product lifecycle policy is described here. You may find a table of products including version numbers and end-of-support dates there. The information is also listed below:

Agent for Windows			
Agent 5.9.1	August 2022	August 2025	August 2027
Agent 5.9.0	April 2022	April 2025	April 2027
Agent 5.8.1	December 2021	December 2024	December 2026
Agent 5.8.0	July 2021	July 2024	July 2026
Agent 3.7.1	December 2020	December 2023	December 2025
Agent 3.7.0	September 2020	September 2023	September 2025
Agent 3.6.1	December 2019	December 2022	December 2024
Agent 3.6.0	August 2019	August 2022	August 2024
Agent 3.5.2	December 2018	December 2021	December 2023
Agent 3.5.1	August 2018	August 2021	August 2023
Agent 3.4.3	December 2017	December 2020	December 2022
Agent 3.4.2	September 2017	September 2020	September 2022
Agent 3.4.1	May 2017	May 2020	May 2022
Agent 3.4.0	February 2017	February 2020	February 2022
Agent 3.3.1 – 3.3.2	May 2016	May 2019	May 2021
Agent 3.3.0	December 2015	December 2018	December 2020
Agent 3.2.3	July 2015	July 2018	July 2020
Agent 3.2.0 – 3.2.2	January 2014	January 2017	January 2019
Agent 3.1.0 – 3.1.3	July 2013	July 2016	July 2018
Agent 3.0.0 – 3.0.1	January 2013	January 2016	January 2018
Authentication Service			
DirectControl 5.9.1	August 2022	August 2025	August 2027
DirectControl 5.9.0	April 2022	April 2025	April 2027
DirectControl 5.8.1	December 2021	December 2024	December 2026

DirectControl 5.8.0	July 2021	July 2024	July 2026
DirectControl 5.7.1	December 2020	December 2023	December 2025
DirectControl 5.7.0	September 2020	September 2023	September 2025
DirectControl 5.6.1	December 2019	December 2022	December 2024
DirectControl 5.6.0	August 2019	August 2022	August 2024
DirectControl 5.5.3	February 2019	February 2022	February 2024
DirectControl 5.5.2	December 2018	December 2021	December 2023
DirectControl 5.5.1	August 2018	August 2021	August 2023
DirectControl 5.5.0	April 2018	April 2021	April 2023
DirectControl 5.4.3	December 2017	December 2020	December 2022
DirectControl 5.4.2	September 2017	September 2020	September 2022
DirectControl 5.4.1	May 2017	May 2020	May 2022
DirectControl 5.4.0	February 2017	February 2020	February 2022
DirectControl 5.3.1	May 2016	May 2019	May 2021
DirectControl 5.3.0	December 2015	December 2018	December 2020
DirectControl 5.2.3	July 2015	July 2018	July 2020
DirectControl 5.2.2	February 2015	February 2018	February 2020
DirectControl 5.2.0 – 5.2.1	August 2014	August 2017	August 2019
DirectControl 5.1.3	January 2014	January 2017	January 2019
DirectControl 5.1.1 – 5.1.2	July 2013	July 2016	July 2018
DirectControl 5.1.0	January 2013	January 2016	January 2018
DirectControl 5.0.5	December 2012	December 2015	December 2017
DirectControl 5.0.4	September 2012	September 2015	September 2017
DirectControl 5.0.0 – 5.0.3	October 2011	October 2014	October 2016
DirectControl 4.4.4	May 2012	May 2015	May 2017
DirectControl 4.4.x	January 2010	January 2013	January 2015
DirectControl 4.3.x	May 2009	May 2012	May 2014
DirectControl 4.2.x	December 2008	December 2011	December 2013

DirectControl 4.x	October 2007	October 2010	October 2012
DirectControl 3.x	April 2006	April 2009	April 2011
DirectControl 2.x	September 2005	September 2008	September 2010
DirectControl 1.x	March 2005	March 2008	March 2010
Privilege Elevation Service			
DirectAuthorize 5.9.1	August 2022	August 2025	August 2027
DirectAuthorize 5.9.0	April 2022	April 2025	April 2027
DirectAuthorize 5.8.1	December 2021	December 2024	December 2026
DirectAuthorize 5.8.0	July 2021	July 2024	July 2026
DirectAuthorize 3.7.1	December 2020	December 2023	December 2025
DirectAuthorize 3.7.0	September 2020	September 2023	September 2025
DirectAuthorize 3.6.1	December 2019	December 2022	December 2024
DirectAuthorize 3.6.0	August 2019	August 2022	August 2024
DirectAuthorize 3.5.2	December 2018	December 2021	December 2023
DirectAuthorize 3.5.1	August 2018	August 2021	August 2023
DirectAuthorize 3.5.0	April 2018	April 2021	April 2023
DirectAuthorize 3.4.3	December 2017	December 2020	December 2022
DirectAuthorize 3.4.2	September 2017	September 2020	September 2022
DirectAuthorize 3.4.1	May 2017	May 2020	May 2022
DirectAuthorize 3.4.0	February 2017	February 2020	February 2022
DirectAuthorize 3.3.1	May 2016	May 2019	May 2021
DirectAuthroize 3.3.0	December 2015	December 2018	December 2020
DirectAuthorize 3.2.3	July 2015	July 2018	July 2020
DirectAuthorize 3.2.0 – 3.2.2	January 2014	January 2017	January 2019
DirectAuthorize 3.1.0 – 3.1.3	July 2013	July 2016	July 2018
DirectAuthorize 3.0.0 – 3.0.1	January 2013	January 2016	January 2018
DirectAuthorize 2.x	October 2011	October 2014	October 2016
DirectAuthroize 1.2.1	May 2012	May 2015	May 2017

DirectAuthorize 1.2	January 2010	January 2013	January 2015
DirectAuthorize 1.x	December 2008	October 2010	October 2012
Audit & Monitoring Service			
DirectAudit 5.9.1	August 2022	August 2025	August 2027
DirectAudit 5.9.0	April 2022	April 2025	April 2027
DirectAudit 5.8.1	December 2021	December 2024	December 2026
DirectAudit 5.8.0	July 2021	July 2024	July 2026
DirectAudit 3.7.1	December 2020	December 2023	December 2025
DirectAudit 3.7.0	September 2020	September 2023	September 2025
DirectAudit 3.6.1	December 2019	December 2022	December 2024
DirectAudit 3.6.0	August 2019	August 2022	August 2024
DirectAudit 3.5.2	December 2018	December 2021	December 2023
DirectAudit 3.5.1	August 2018	August 2021	August 2023
DirectAudit 3.5.0	April 2018	April 2021	April 2023
DirectAudit 3.4.3	December 2017	December 2020	December 2022
DirectAudit 3.4.2	September 2017	September 2020	September 2022
DirectAudit 3.4.1	May 2017	May 2020	May 2022
DirectAudit 3.4.0	February 2017	February 2020	February 2022
DirectAudit 3.3.1	May 2016	May 2019	May 2021
DirectAudit 3.3.0	December 2015	December 2018	December 2020
DirectAudit 3.2.3	July 2015	July 2018	July 2020
DirectAudit 3.2.0 – 3.2.2	January 2014	January 2017	January 2019
DirectAudit 3.1.0 – 3.1.3	July 2013	July 2016	July 2018
DirectAudit 3.0.0 – 3.0.1	January 2013	January 2016	January 2018
DirectAudit 2.x	October 2011	October 2014	October 2016
DirectAudit 1.3.x	February 2011	February 2014	February 2016
DirectAudit 1.1.x	July 2008	July 2011	July 2013
DirectAudit 1.x	May 2007	May 2010	May 2012

Isolation & Encryption Service			
DirectSecure 5.4.2	October 2017	October 2020	October 2022
DirectSecure 5.4.0	March 2017	March 2020	March 2022
DirectSecure 5.3.1	August 2016	August 2019	August 2021
DirectSecure 5.2.3	August 2015	August 2018	August 2020
DirectSecure 5.2.2	May 2015	May 2018	May 2020
DirectSecure 5.1.1	August 2013	August 2016	August 2018
DirectSecure 1.2.x	August 2011	August 2014	August 2016
DirectSecure 1.x	May 2009	May 2012	May 2014
Deployment Manager			
Deployment Manager 5.5.2	December 2018	December 2021	December 2023
Deployment Manager 5.5.1	August 2018	August 2021	August 2023
Deployment Manager 5.5.0	April 2018	April 2021	April 2023
Deployment Manager 5.4.3	December 2017	December 2020	December 2022
Deployment Manager 5.4.2	September 2017	September 2020	September 2022
Deployment Manager 5.4.1	May 2017	May 2020	May 2022
Deployment Manager 5.4.0	February 2017	February 2020	February 2022
Deployment Manager 5.3.1	May 2016	May 2019	May 2021
Deployment Manager 5.3.0	December 2015	December 2018	December 2020
Deployment Manager 5.2.3	July 2015	July 2018	July 2020
Deployment Manager 5.2.2	February 2015	February 2018	February 2020
Deployment Manager 5.2.0	August 2014	August 2017	August 2019
Deployment Manager 5.1.3	January 2014	January 2017	January 2019
Deployment Manager 5.1.0 – 5.1.2	January 2013	January 2016	January 2018
Deployment Manager 2.x	May 2011	May 2014	May 2016
Deployment Manager 1.x	June 2010	June 2013	June 2015
Zone Provisioning Agent			
ZPA 5.9.1	August 2022	August 2025	August 2027

ZPA 5.9.0	April 2022	April 2025	April 2027
ZPA 5.8.1	December 2021	December 2024	December 2026
ZPA 5.8.0	July 2021	July 2024	July 2026
ZPA 5.7.1	December 2020	December 2023	December 2025
ZPA 5.7.0	September 2020	September 2023	September 2025
ZPA 5.6.1	December 2019	December 2022	December 2024
ZPA 5.6.0	August 2019	August 2022	August 2024
ZPA 5.5.2	December 2018	December 2021	December 2023
ZPA 5.5.1	August 2018	August 2021	August 2023
ZPA 5.5.0	April 2018	April 2021	April 2023
ZPA 5.4.3	December 2017	December 2020	December 2022
ZPA 5.4.2	September 2017	September 2020	September 2022
ZPA 5.4.1	May 2017	May 2020	May 2022
ZPA 5.4.0	February 2017	February 2020	February 2022
ZPA 5.3.1	May 2016	May 2019	May 2021
ZPA 5.3.0	December 2015	December 2018	December 2020
ZPA 5.2.3	July 2015	July 2018	July 2020
ZPA 5.2.2	February 2015	February 2018	February 2020
ZPA 5.2.0	August 2014	August 2017	August 2020
ZPA 5.1.3	January 2014	January 2017	January 2019
ZPA 5.1.1 – 5.1.2	July 2013	July 2016	July 2018
ZPA 5.1.0	January 2013	January 2016	January 2018
ZPA 5.0.x	August 2011	August 2014	August 2016
ZPA 1.x	February 2010	February 2013	February 2015
Delinea OpenSSH			
OpenSSH 5.9.1	August 2022	August 2025	August 2027
OpenSSH 5.9.0	April 2022	April 2025	April 2027
OpenSSH 5.8.1	December 2021	December 2024	December 2026

OpenSSH 5.8.0	July 2021	July 2024	July 2026
OpenSSH 5.7.1	December 2020	December 2023	December 2025
OpenSSH 5.7.0	September 2020	September 2023	September 2025
OpenSSH 5.6.1	December 2019	December 2022	December 2024
OpenSSH 5.6.0	August 2019	August 2022	August 2024
OpenSSH 5.5.2	December 2018	December 2021	December 2023
OpenSSH 5.5.1	August 2018	August 2021	August 2023
OpenSSH 5.5.0	April 2018	April 2021	April 2023
OpenSSH 5.4.3	December 2017	December 2020	December 2022
OpenSSH 5.4.2	September 2017	September 2020	September 2022
OpenSSH 5.4.1	May 2017	May 2020	May 2022
OpenSSH 5.4.0	February 2017	February 2020	February 2022
OpenSSH 5.3.1	May 2016	May 2019	May 2021
OpenSSH 5.3.0	December 2015	December 2018	December 2020
OpenSSH 5.2.3	July 2015	July 2018	July 2020
OpenSSH 5.2.2	February 2015	February 2018	February 2020
OpenSSH 5.1.3 – 5.1.4	January 2014	January 2017	January 2019
OpenSSH 5.1.1 – 5.1.2	July 2013	July 2016	July 2018
OpenSSH 5.1.0	January 2013	January 2016	January 2018
OpenSSH 4.5.5	December 2012	December 2015	December 2017
OpenSSH 4.5.4	September 2012	September 2015	September 2017
OpenSSH 4.5.x	February 2011	February 2014	February 2016
OpenSSH 4.3.x	January 2010	January 2013	January 2015
OpenSSH 4.1.x	September 2008	September 2011	September 2013
OpenSSH 3.x	July 2006	July 2009	July 2011
Delinea for Samba			
Samba Integration 5.9.0	April 2022	April 2025	April 2027
Samba Integration 5.7.0	October 2020	October 2023	October 2025

Samba Integration 5.5.2	January 2019	January 2022	January 2024
Samba Integration 5.5.0	June 2018	June 2021	June 2023
Samba Integration 5.4.3	February 2018	February 2021	February 2023
Samba Integration 5.4.0	May 2017	May 2020	May 2022
Samba Integration 5.3.0	May 2016	May 2019	May 2021
Delinea for DB2			
DirectControl for DB2 5.8.0	July 2021	July 2024	July 2026
DirectControl for DB2 5.7.0	September 2020	September 2023	September 2025
DirectControl for DB2 5.4.0	May 2017	May 2020	May 2022
DirectControl for DB2 5.2.3	August 2015	August 2018	August 2020
DirectControl for DB2 4.5.0	January 2015	January 2018	January 2020
DirectControl for DB2 4.4.4	August 2012	August 2015	August 2017
DirectControl for DB2 4.1.x	June 2008	June 2011	June 2013
Delinea PuTTY			
PuTTY 5.x-0.76	December 2021	December 2024	December 2026
PuTTY 5.x-0.74	July 2021	July 2024	July 2026
PuTTY 5.x-0.73	September 2020	September 2023	September 2025
PuTTY 5.x-0.71	August 2019	August 2022	August 2024
PuTTY 5.x-0.70	April 2018	April 2021	April 2023
PuTTY 5.x-0.69	September 2017	September 2020	September 2022
PuTTY 5.x-0.67	February 2017	February 2020	February 2022
PuTTY 5.x-0.64	July 2015	July 2018	July 2020
PuTTY 5.x-0.63	November 2013	November 2016	November 2018
PuTTY 5.x-0.62	July 2013	July 2016	July 2018
PuTTY 3.x-0.60	October 2007	October 2010	October 2012
PuTTY 3.x-0.59	July 2006	July 2009	July 2011
Delinea Kerberos Tools			
Tools 5.1.0	March 2013	March 2016	March 2018

Tools 4.x	February 2009	February 2012	February 2014
Tools 3.x	July 2006	July 2009	July 2011

Note:

- Deployment Manager, and Delinea Kerberos Tools, are discontinued and hence no more new releases are available. (Ref: CS-47626)
- DirectSecure was deprecated in release 2020. The last supported version of DirectSecure was version 5.4.2 (Release 2017.2). No new release of DirectSecure is available after the 5.4.2 (2017.2) version.

Server Suite 2022.1 Release Notes

The DelineaServer Suite (previously called Centrify Infrastructure Services, or Centrify Zero Trust Privilege Services) is an integrated family of directory-based authentication, privileged access, privileged elevation, audit & monitoring solutions that secure your cross-platform environment and strengthen regulatory compliance initiatives.

Server Suite includes the following components:

- Authentication Service secures your platforms using the same authentication and Group Policy services deployed for your Windows environment.
- Privilege Elevation Service centrally manages and enforces role-based entitlements for fine-grained control of user access and privileges on UNIX, Linux, and Windows systems.
- Audit & Monitoring Service delivers auditing, logging, and real-time monitoring of user activity on your Windows, UNIX, and Linux system.

This integrated solution helps you improve IT efficiency, strengthen regulatory compliance initiatives, and centrally secure your heterogeneous computing environment.

This release notes cover information specifically about Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service.

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

This release usually includes packages for Windows, UNIX, and Linux operating system environments.

The files for this release are organized onto two media, each available in ISO and zip form:

Server Suite for 64-bit Windows

- main folder

This is the main folder containing information pertinent to this release.

- The readme.txt file provides a summary of where to find files in a plain text format.
- Copyright.txt and Acknowledgements.txt provide copyright information and legal notices for third party and open-source software used in Delinea Server Suite.
- Delinea-end-user-license-agreement.txt provides the text of the license agreement displayed during installation.
- autorun.inf controls the autorun program, autorun.exe, on Windows computers.

The following are sub-folders that are organized to provide you access to different software components in the Delinea Server Suite.

- Agent folder

  This folder contains the installer packages for installing Server Suite Agent for Windows on Windows computers.

- Common folder

  This folder contains the installer packages for common components necessary for all Delinea products on Windows computers.

- DirectAudit folder

  This folder contains the installer packages for Delinea Audit & Monitoring Service on Windows computers.

- DirectManage folder

  This folder contains the installer packages for Delinea Authentication Service and Delinea Privilege Elevation Service on Windows computers.

- LicensingService Folder

This folder contains the installer packages for Delinea Licensing Service utilities on Windows computers.

- Resources Folder

  This folder contains resources for internal use for the media. It can be safely ignored.

Server Suite Agents for UNIX/Linux

This image contains a zipped bundle of files for Server Suite agent on each supported UNIX, or Linux platform and an adcheck utility for each supported platform.

You may find the appropriate bundle for an OS platform based on the following table:

delinea-server-suite-<release number>-aix7.1-ppc.tgz	IBM AIX, IBM VIOS
delinea-server-suite-<release number>-cos-x86_64.tgz	Flatcar, RHCOS
delinea-server-suite-<release number>-deb9-arm64.tgz	Ubuntu
delinea-server-suite-<release number>-deb9-ppc64el.tgz	Ubuntu
delinea-server-suite-<release number>-deb9-x86_64.tgz	Debian, Ubuntu
delinea-server-suite-<release number>-hp11.31-ia64.tgz	HPUX
delinea-server-suite-<release number>-hp11.31-pa.tgz	HPUX
delinea-directcontrol-<release number>-mac10.15.tgz	MAC (Intel, M1)
delinea-server-suite-<release number>-rhel6-ppc64.tgz	RHEL
delinea-server-suite-<release number>-rhel6-x86_64.tgz	Amazon Linux, CentOS, Fedora, Oracle Linux, RHEL, AlmaLinux, Rocky Linux
delinea-server-suite-<release number>-rhel7-aarch64.tgz	Amazon Linux, CentOS, Oracle Linux, RHEL
delinea-server-suite-<release number>-rhel7-ppc64le.tgz	RHEL
delinea-server-suite-<release number>-sol10-sparc.tgz	Oracle Solaris
delinea-server-suite-<release number>-sol10-x86.tgz	Oracle Solaris
delinea-server-suite-<release number>-sol11-i386.tgz	Oracle Solaris (IPS package)
delinea-server-suite-<release number>-sol11-sparc.tgz	Oracle Solaris (IPS package)
delinea-server-suite-<release number>-suse12-aarch64.tgz	SUSE
delinea-server-suite-<release number>-suse12-ppc64le.tgz	SUSE
delinea-server-suite-<release number>-suse12-x86_64.tgz	SUSE

Notes:

- The OS version number specified in the bundle name indicates the minimum OS version that it supports.
- You should also choose the appropriate bundle for the specific architecture as indicated in the bundle name.
- Inside each bundle, it contains packages of associated products supported on that platform. The naming convention follows the above bundle names except that the prefix of a package reflects the product it serves. The following are the possible package prefixes and the corresponding product

names:

CentrifyDA	Delinea DirectAudit package
CentrifyDC	Delinea DirectControl package
CentrifyDC-cifsidmap	Delinea for CIFS ID mapping package
CentrifyDC-curl	Required component of Delinea DirectControl package
CentrifyDC-ldapproxy	Delinea OpenLDAP Proxy package
CentrifyDC-nis	Delinea Network Information Service and Delinea NIS Server package
CentrifyDC-openldap	Required component of Delinea DirectControl package
CentrifyDC-openssh	Delinea OpenSSH package
CentrifyDC-openssl	Required component of Delinea DirectControl package

- Before installation, please review the *Upgrade and Compatibility Guide* and run the adcheck utility to make sure the environment is ready, especially if you are using native package manager to install.

Go to Supported Versions.

Newly Added Supported Platforms

- RHEL 9
- RHEL 8.6
- Ubuntu 22.04

Supported UNIX/Linux Platforms

AlmaLinux 8.5	x86_64	Yes	Yes	Yes	
Alpine Linux 3.13, 3.14	X86_64	No	Yes	Yes	
Amazon Linux 2 LTS	aarch64	No	Yes	Yes	
Amazon Linux 2 LTS	x86_64	No	Yes	Yes	
CentOS 7.4-7.9, 8.0-8.5	aarch64	No	Yes	Yes	
CentOS 6.0-6.10, 7.0-7.9, 8.0-8.5	x86_64	Yes	Yes	Yes	
Debian 9.0-9.13, 10.0-10.11, 11	x86_64	Yes	Yes	Yes	
Flatcar	x86_64	No	Yes	Yes	

HP-UX 11.31 (Trusted and Untrusted)	Itanium	No	Yes	Yes	
HP-UX 11.31 (Trusted and Untrusted)	PA-RISC	No	Yes	Yes	
IBM AIX 7.1 TL1+, 7.2	ppc	No	Yes	Yes	Note 4
IBM Virtual I/O Server 3.x	ppc	No	Yes	Yes	
MacOS 11.0-11.6, 12	M1	No	Yes	No	Note 3
MacOS 10.15, 11.0-11.6, 12	x86_64	No	Yes	No	Note 3
Oracle Linux 7.4-7.9, 8.0-8.5	aarch64	No	Yes	Yes	
Oracle Linux 6.0-6.10, 7.0-7.9, 8.0-8.5	x86_64	Yes	Yes	Yes	
Oracle Solaris 10 u8+, 11.0-11.4	SPARC	No	Yes	Yes	Note 2
Oracle Solaris 10 u8+, 11.0-11.4	x86_64	No	Yes	Yes	Note 2
Red Hat Enterprise Linux 7.4-7.9, 8.0-8.6, 9	aarch64	No	Yes	Yes	
Red Hat Enterprise Linux 6.0-6.10, 7.0-7.9	ppc64	Yes	Yes	Yes	
Red Hat Enterprise Linux 7.1-7.9, 8.0-8.6, 9	ppc64le	Yes	Yes	Yes	
Red Hat Enterprise Linux 8.0	S390	No	Yes	No	
Red Hat Enterprise Linux 6.0-6.10, 7.0-7.9, 8.0-8.6, 9	x86_64	Yes	Yes	Yes	
Red Hat Enterprise Linux CoreOS (RHCOS)	x86_64	Yes	Yes	Yes	Note 1
Red Hat Fedora Linux 35	x86_64	Yes	Yes	Yes	
Rocky Linux 8.5	x86_64	Yes	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	aarch64	No	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	ppc64le	Yes	Yes	Yes	
SUSE Enterprise Linux 12 SP4	S390	No	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	x86_64	Yes	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 22.04	arm64	No	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 22.04	ppc64el	Yes	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 22.04	x86_64	Yes	Yes	Yes	

Note 1: Please refer to the *Planning and Deployment Guide* for features supported on this platform.

Note 2: Starting with Release 2020, we require the OS patch level update 8 or above on Solaris 10.

Note 3: Delinea OpenSSH is not supported on this platform.

Note 4: Starting with Release 2021.1, we require the TL1 or above on AIX 7.1.

Additional Information

You should follow the OS vendors' recommendation to update the necessary patches. Here are the minimum patch requirements for the specific UNIX platforms (Ref: CS-45562):

1. HPUX 11.31
    1. PHNE_40225 - Cumulative Console and BSD Pty Patch (it is required for DirectAudit package)
2. Solaris 10 x86_64
    1. 119255-66
    2. 127128-11
    3. 141445-09
    4. 142910-17
3. Solaris 10 SPARC
    1. 119254-66
    2. 120011-14
    3. 127127-11
    4. 142909-17

Supported Windows Platforms

The following 64-bit Windows platforms are supported on Delinea Server Suite (Ref: CS-49379):

- Windows 10 LTSB/LTSC (Note 1)
- Windows 11 LTSB/LTSC
- Windows Server 2012
- Windows Server 2012R2
- Windows Server 2016
- Windows Server 2019 LTSC
- Windows Server 2022 LTSC
- Windows Server 2012 Core (Note 2)
- Windows Server 2012 Minimum Server Interface (Note 2)
- Windows Server 2012R2 Core (Note 2)
- Windows Server 2012R2 Minimum Server Interface (Note 2)

Note:

1. We support Windows 10 Long Term Servicing Channel (LTSC), or previously called Long Term Servicing Branch (LTSB), editions based on Microsoft's lifecycle fact sheet https://docs.microsoft.com/en-us/lifecycle/faq/windows and https://docs.microsoft.com/en-us/windows/release-health/release-information
2. Only the Privilege Elevation Service component of Server Suite Agent for Windows supports these platforms (Core and Minimum Server Interface)
3. Support for all 32-bit Windows platform was terminated in 2015 (Server Suite 2015.1)

Also note that Server Suite require specific versions of .NET to work. Please refer to the following table for the requirement (Ref: CS-49381):

Server Suite 2022.1	August 2022	4.8	--*
Server Suite 2022	April 2022	4.8	--*
Server Suite 2021.1	December 2021	4.8	--*
Server Suite 2021	July 2021	4.8	--*
Infrastructure Services 2020.1	December 2020	4.6.2	--*
Infrastructure Services 2020	September 2020	4.6.2	--*

Infrastructure Services 19.9	December 2019	4.6.2	--*
Infrastructure Services 19.6	August 2019	4.6.2	--*
Infrastructure Services 18.11	December 2018	4.6.2	4.6.2
Infrastructure Services 18.8	August 2018	4.6.2	4.6.2
Infrastructure Services 2018	April 2018	4.6.2	4.6.2
Infrastructure Services 2017.3	December 2017	4.5.2	4.5.2
Infrastructure Services 2017.2	September 2017	4.5.2	4.5.2
Server Suite 2017.1	May 2017	4.5	4.5.2
Server Suite 2017	February 2017	4.5	4.5.2
Server Suite 2016.1	May 2016	4.5	4.5.2
Server Suite 2016	December 2016	4.5	4.5.2

Note: We no longer bundle .NET in our installation media any more starting Release 19.6. (Ref: CS-47940)

This is the last release to support the following operating system platforms:

- macOS
- HP-UX 11.31 (PA-RISC)

Delinea has established product security policies documented at our support page. You may also find the details of all the published security advisories there.

For component specific security fixes in this release, you may find them in the corresponding component release-notes.html files. Please refer to Section 7 for a description of individual release notes.

See Component Version Table.

- For Access Manager, DirectControl agent and Delinea OpenSSH, see the Authentication Service and Privilege Elevation Service Release Notes.
- For Audit Manager and DirectAudit agent, see the Audit & Monitoring Service Release Notes.
- For Agent for Windows, see the Agent for Windows Release Notes.
- For Server Suite PuTTY, see the Server Suite PuTTY Release Notes

  Also, see Product Lifecycle Versions for product versions.

You can get all the supported releases from the download center in Delinea support web site.

- All the ISO, ZIP, and TGZ files are associated with the MD5 checksum.
- All RPM and DEB packages as well as YUM and APT repositories are also protected by the GPG signature. You can find the GPG public key in the download center.

Component specific bug fixes in this release can be found in the corresponding component release-notes.html files. Please refer to Release Notes for Server Suite Components for a description of individual release notes.

Component specific known issues/limitations can be found in the corresponding component release notes files. Please refer to Release Notes for Server Suite Components for a description of individual release notes.

For the most up to date list of known issues, please login to the Customer Support Portal at https://www.delinea.com/support and refer to Knowledge Base articles for any known issues with the release.

In addition to the documentation provided with this package, you can find the answers to common questions and information about any general or platform-specific known limitations as well as tips and suggestions from the Delinea Knowledge Base.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

- 2022.1 Auth/PE Release Notes
- 2022 Auth/PE Release Notes

About this Release

Authentication Service and Privilege Elevation Service, part of the product category Delinea Server Suite (previously called Centrify Infrastructure Services or Centrify Zero Trust Privilege Services), centralize authentication and privileged user access across disparate systems and applications by extending Active Directory-based authentication, enabling use of Windows Group Policy and Single-Sign-On. With Delinea Server Suite, enterprises can easily migrate and manage complex UNIX, Linux, and Windows systems, rapidly consolidate identities into the directory, organize granular access and simplify administration. Delinea Authentication Service, through Delinea's patented Zone technology, allows organizations to easily establish global UNIX identities, centrally manage exceptions on Legacy systems, separate identity from access management and delegate administration. Delinea's non-intrusive and organized approach to identity and access management results in stronger security, improved compliance and reduced operational costs.

The Upgrade Guide describes the correct order to perform updates such that all packages continue to perform correctly once upgraded.

The product-related release notes and documents are available online at https://docs.delinea.com/.

Feature Changes in this Release

For a list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite Release Notes.

For a list of platforms that Delinea will remove support in upcoming releases, refer to the 'Notice of Termination Support' section in the Server Suite Release Notes).

- Implemented a new feature to gather statistic information for NSS requests. See Configuration Parameters for details.
- Implemented a new feature to write info and warning logs when the time spent on a complete NSS request exceeds the configured threshold value. See Configuration Parameters for details.

General

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Security Fix

Server Suite DirectControl Agent for

- Added support for a systemd environment file under the /etc/default directory. (Ref: 394029)
- If the domain controller has installed Windows updates dated November 9, 2021 or later and set the new registry value "PacRequestorEnforcement" as "2", resetting passwords via Kerberos would fail. As a result, DirectControl adjoin, adkeytab, and adpasswd command line utilities would fail to reset accounts' password. Microsoft has confirmed this issue, please see the Known issues section of this article: https://prod.support.services.microsoft.com/en-us/topic/kb5008380-authentication-updates-cve-2021-42287-9dafac11-e0d0-4cb8-959a-143bd0201041 (Ref:430402)

  This release has a solution to bypass that issue.

DirectControl Command Line Utilities

Configuration Parameters

New Parameters

New Parameters for DirectControl

- pam.homedir.update.ownership: false This parameter specifies whether or not to update the home directory ownership when the user logs in. The default is false. (Ref: 430517)
- adclient.nss.statistic.interval: 30m This parameter specifies the statistic interval for adclient to gather NSS query statistics information. The default is 30

minutes. (Ref: 433831)

Added the following Server Suite configuration items for the NSS module:

Below are the global settings for all categories of NSS requests (in milliseconds): nss.watch.slow.lookup.info.threshold: -1 nss.watch.slow.lookup.warn.threshold: -1

Below are the per-category settings, append "user" or "group" suffix, which can override global settings. "user" category indicates these NSS calls: getpwnam* getpwuid* getgrouplist "group" category indicates these NSS calls: getgrnam* getgrgid*

nss.watch.slow.lookup.info.threshold.user: -1 nss.watch.slow.lookup.warn.threshold.user: -1 nss.watch.slow.lookup.info.threshold.group: -1 nss.watch.slow.lookup.warn.threshold.group: -1

Added the following Server Suite configuration items for adclient: Below are the global settings for all categories of NSS requests (in milliseconds): adclient.watch.slow.lookup.info.threshold: -1 adclient.watch.slow.lookup.warn.threshold: -1

Below are the per-category settings, append "user" or "group" suffix, which can override global settings. "user" category indicates these NSS calls: getpwnam* getpwuid* getgrouplist "group" category indicates these NSS calls: getgrnam* getgrgid*

adclient.watch.slow.lookup.info.threshold.user: -1 adclient.watch.slow.lookup.warn.threshold.user: -1 adclient.watch.slow.lookup.info.threshold.group: -1 adclient.watch.slow.lookup.warn.threshold.group: -1

New Parameters for OpenLDAP Proxy

- Added the support of bypassing caches for specified categories with the following new parameters in slapd.conf:
  - ldapproxy.bypass.adclientcache: set this parameter to specify some categories (separating with comma; "*" means all searches) to enable this feature.
  - ldapproxy.bypass.slapdcache: set this parameter in the /etc/centrifydc/openldap/ldapproxy.slapd.conf file to specify some categories (separating with comma; "*" means all searches) to enable this feature.
    Note: USER and GROUP categories always use caches.

Modified Parameters

Audit Trail Events

Server Suite Access Manager

Server Suite Access Module for PowerShell

Server Suite Group Policy Management

Server Suite Licensing Service

Server Suite OpenLDAP Proxy

Server Suite OpenSSH

Server Suite OpenSSL

Server Suite Report Services

- You can now specify registry keys to be created during the Report Services silent installation so you don't need to go back after it's installed to specify additional config parameters. (Ref:430713)

Server Suite Smart Card

- Added Smart Card Support for Rocky Linux. (Ref:433531)
- Added Smart Card Support for Alma Linux. (Ref:433532)

Server Suite Windows Installer

Server Suite Windows SDK

Server Suite Zone Provisioning Agent

Fixed Issues in this Release

General

- We have fixed the memory allocation issues related to Microsoft KB: KB5014697 (Win 11) / KB5014692 (Win10) / KB5014699 (Win2019) / KB5014702 (Win2016) KB updates. (Ref:441208)

Security Fixes

- Fixed the high severity CVE-2022-37434 of zlib with the official patch. (Ref:454508)
- OpenSSL was upgraded from 3.0.1. to 3.0.5 (Ref: 444612)
- Centrify cURL was upgraded based on cURL v7.84.0.(Ref:442043)
- Upgraded zlib to 1.2.12.(Ref:430916)

Server Suite DirectControl Agent for

- The obsolete group policy script TestFipsMode.pl has been removed from the CentrifyDC package. (Ref: 427705)
- Fixed an issue where single sign-on could fail when using the KCM kerberos credential cache. (Ref: 430385)
- Fixed an issue where dzdo may crash on Debian if audit is disabled in a kernel parameter. (Ref: 442864)
- Fixed an issue where adjoin didn't write Kerberos keytab entries of the computer userPrincipalName. (Ref: 442592)
- Fixed an issue where some groups may have lost members when 'adclient.local.group.merge' was true. (Ref: 431082)
- Fixed an issue where centrifydc.log was empty after log rotation on RHEL 8 (Ref: 429155)
- Fixed a race condition issue that would sometimes crash processes performing NSS user lookups. (Ref:443314)
- Fixed an issue where the user needed to add '+' at the end of /etc/passwd manually to resolve the AD user while NSS compatability mode was enabled. (Ref:441847)
- Fixed an issue where user lookup by NTLM name would fail when the adclient.included.domains setting was in place.(Ref:422590)

DirectControl Command Line Utilities

DirectControl Installation

Audit Trail Events

Server Suite Access Manager

- Fixed an issue where Access Manager showed an unknown OS type for AlmaLinux, Rocky Linux, Red Hat Enterprise Linux CoreOS, and Flatcar Container Linux. (Ref: 431687)
- Fixed an issue in Access Manager where users couldn't change the license container in the zone's properties dialog box. (Ref: 433612)
- Fixed an issue with the Access Manager analyzer where it showed incorrect orphaned role assignments. (Ref: 431820)

Server Suite Access Module for PowerShell

Server Suite ADEdit

- Fixed an issue where the adedit command 'create_assignment' couldn't create role assignments for the same user but with different start/end time. (Ref:422588)

Server Suite Group Policy Management

Server Suite Licensing Service

Server Suite NIS

Server Suite OpenLDAP Proxy

- Fixed an issue where the ldapproxy in-memory cache couldn't work with sizelimit. (Ref:445830)
- Fixed an issue where ldapproxy replied with two searchResultDone packets for one paged search.(Ref:442434)
- Fixed an issue where ldapproxy returned "no such object" when searching for an rfc2307nismap container. (Ref:441860)

Server Suite OpenSSH

Server Suite Report Services

- Fixed an issue where Report Services ended prematurely because of a null exception. (Ref: 430704)

Server Suite Smart Card

Server Suite Windows Installer

Server Suite Windows SDK

Server Suite Zone Provisioning Agent

Fixes in Release 2022.1 Component Update

Fixed several critical security fixes by upgrading 'Centrify OpenSSL' to 3.0.7. (Ref: 469895)

Fixed issues related to CVE-2022-42915 by upgrading cURL to 7.86.0. (Ref: 469896)

Known Issues

The following sections describe common limitations or known issues associated with this Authentication Service and Privilege Elevation Service release.

For the most up to date list of known issues, please login to the Customer Support Portal at https://www.delinea.com/support and refer to Knowledge Base articles for any known issues with the release.

Server Suite DirectControl Agent for

- Known Issues with Multi-Factor Authentication (MFA)

  If MFA is enabled but the parameter "adclient.legacyzone.mfa.required.groups" is set to a non-existent group, all AD users will be required for MFA. The workaround is to remove any non-existent groups from the parameter. (Ref: CS-39591b)

- Known Issues with AIX

  On AIX, upgrading DirectControl agent from 5.0.2 or older versions in disconnected mode may cause unexpected behavior. The centrifydc service may be down after upgrade. It's recommended not to upgrade DirectControl agent in disconnected mode. (Ref: CS-30494a)

  Some versions of AIX cannot handle username longer than eight characters. As a preventive measure, we have added a new test case in the adcheck command to check if the parameter LOGIN_NAME_MAX is set to 9. If yes, adcheck will show a warning so that users can be aware of it. (Ref: CS-30789a)

- Known issues with Fedora 19 and above (Ref: CS-31549a, CS-31730a)

  There are several potential issues on Fedora 19 and above:

  1. The adcheck command will fail if the machine does not have Perl installed.
  2. Group Policy will not be fully functional unless Text/ParseWords.pm is installed.

- Known issues with RedHat

When logging into a RedHat system using an Active Directory user that has the same name as a local user, the system will not warn the user of the conflict, which will result in unpredictable login behavior. The workaround is to remove the conflict or login with a different AD user. (Ref: CS-28940a, CS-28941a)

  - Known issues with rsh / rlogin (Ref: IN-90001)
  - When using rsh or rlogin to access a computer that has DirectControl agent installed, and where the user is required to change their password, users are prompted to change their password twice. Users may use the same password each time they are prompted, and the password is successfully changed.

- Known issues with compatibility

  Using DirectControl 4.x agents with Access Manager 5.x (Ref: IN-90001)

  - DirectControl 4.x agents can join classic zones created by Access Manager 5.x. It will ostensibly be able to join a DirectControl 4.x agent to a hierarchical zone as well, but this causes failure later as such behavior is undefined.

  Default zone not used in DirectControl 5.x (Ref: IN-90001)

  - In DirectControl 4.x, and earlier, there was a concept of the default zone. When Access Manager was installed, a special zone could be created as the default zone. If no zone was specified when joining a domain with adjoin, the default zone would be used.
  - This concept has been removed from DirectControl 5.0.0 and later as it is no longer relevant with hierarchical zones. In zoned mode, a zone must now always be specified.
  - A zone called "default" may be created, and default zones created in earlier versions of Access Manager may be used, but the name must be explicitly used.

Smart Card

- Release 18.8 includes an update to Coolkey to support Giesecke & Devrient 144k, Gemalto DLGX4-A 144, and HID Crescendo 144K FIPS cards. However, this has caused known issues that may cause CAC cards to only work sporadically. A workaround for CAC cards is to wait for it to prompt for PIN and Welcome, without removing the card, and then try again. (Ref: CC-58013a)
- There is a Red Hat Linux desktop selection issue found in RHEL 7 with smart card login. When login with smart card, if both GNOME and KDE desktops are installed, user can only log into GNOME desktop even though "KDE Plasma Workspace" option is selected. (Ref: CS-35125a)
- On RHEL 5.10 and 5.11, if "Smart Card Support" is enabled and a smartcard is inserted on the login screen, a PIN prompt may not show up until you hit the "Enter" key. The workaround is to replace libsoftokn3.so with the old one on RHEL 5.9, which is a shared object file in NSS package. (Ref: CS-35038a)
- On RHEL 5.10 and 5.11, if "Smart Card Support" is enabled and "Card Removal Action" is configured as "Lock", the screen will be locked several seconds after login with smart card. The workaround is to replace libsoftokn3.so with the old one on RHEL 5.9, which is a shared object file in NSS package. (Ref: CS-33871a)
- When a SmartCard user attempts to login on Red Hat 6.0 with a password that has expired, the authentication error message may not mention that authentication has failed due to an expired password. (Ref: CS-28305a)
- On RedHat, any SmartCard user will get a PIN prompt even if he's not zoned, even though the login attempt will ultimately fail. This is a divergence from Mac behavior - On Mac, if a SmartCard user is not zoned, Mac doesn't even prompt the user for PIN. (Ref: CS-33175c)
- If a SmartCard user's Active Directory password expires while in disconnected mode, the user may still be able to log into their machine using their expired password. This is not a usual case, as secure SmartCard AD environments usually do not allow both PIN and Password logins while using a Smart Card. (Ref: CS-28926a)
- To login successfully in disconnected mode (Ref: CS-29111a):
  - For a password user:
    - A password user must log in successfully once in connected mode prior to logging in using disconnected mode. (This is consistent with other DirectControl agent for *NIX behavior)
  - For a SmartCard user:
    - The above is not true of SmartCard login. Given a properly configured RedHat system with valid certificate trust chain and CRL set up, a SmartCard user may successfully login using disconnected mode even without prior successful logins in connected mode.
    - If certificate trust chain is not configured properly on the RedHat system, the SmartCard user's login attempt will fail.
    - If the SmartCard user's login certificate has been revoked, and the RedHat system has a valid CRL that includes this certificate, then the system will reject the user.

- After upgrading from DirectControl version 5.0.4 to version 5.1, a Smartcard user may not be able to login successfully. The workaround is to run the following CLI commands:

  sudo rm /etc/pam_pkcs11/cacerts/*

  sudo rm /etc/pam_pkcs11/crls/*

  sudo rm /var/centrify/net/certs/*

  then run adgpupdate. (Ref: CS-30025c)

- When CRL check is set via Group Policy and attempting to authenticate via Smartcard, authentication may fail. The workaround is to wait until the Group Policy Update interval has occurred and try again or to force an immediate Group Policy update by running the CLI command adgpupdate. (Ref: CS-30090c)

- After upgrading from DirectControl agent Version 5.0.4 to version 5.1.1, a SmartCard user may not be able to authenticate successfully. The workaround is to perform the following CLI command sequence:

```
sctool -d
sctool -e
sudo rm /etc/pam_pkcs11/cacerts/*
sudo rm /etc/pam_pkcs11/crls/*
sudo rm /var/centrify/net/certs/*"
adgpupdate
```

and then re-login using the SmartCard and PIN. (Ref: CS-30353c)

- A name-mapping user can unlock screen with password even though the previous login was with PIN. (Ref: CS-31364b)
- Need to input PIN twice to login using CAC card with PIN on RedHat. It will fail on the first input but succeed on the second one. (Ref: CS-30551c)
- Running "sctool –D" with normal user will provide wrong CRL check result. The work-around is to run it as root. (Ref: CS-31357b)
- Screen saver shows password not PIN prompt (Ref: CS-31559a)

Most smart card users can log on with a smart card and PIN only and cannot authenticate with a username and password. However, it is possible to configure users for both smart card/PIN and username/password authentication. Generally, this set up works seamlessly: the user either enters a username and password at the log on prompt, or inserts a smart card and enters a PIN at the prompt.

However, for multi-user cards, it can be problematic when the screen locks and the card is in the reader. When a user attempts to unlock the screen, the system prompts for a password, not for a PIN, although the PIN is required because the card is in the reader. If the user is not aware that the card is still in the reader and enters his password multiple times, the card will lock once the limit for incorrect entries is reached.

On RHEL 7, an authenticated Active Directory user via smart card cannot login again if the smart card is removed. This is due to a bug in RHEL 7, https://bugzilla.redhat.com/show_bug.cgi?id=1238342. This problem does not happen on RHEL6. (Ref: CSSSUP-6914c)

Report Services

- N/A

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

About this Release

Authentication Service and Privilege Elevation Service, part of the product category Delinea Server Suite (previously called Centrify Infrastructure Services or Centrify Zero Trust Privilege Services), centralize authentication and privileged user access across disparate systems and applications by extending Active Directory-based authentication, enabling use of Windows Group Policy and Single-Sign-On. With Delinea Server Suite, enterprises can easily migrate and manage complex UNIX, Linux, and Windows systems, rapidly consolidate identities into the directory, organize granular access and simplify administration. Delinea Authentication Service, through Delinea's patented Zone technology, allows organizations to easily establish global UNIX identities, centrally manage exceptions on Legacy systems, separate identity from access management and delegate administration. Delinea's non-intrusive and organized approach to identity and access management results in stronger security, improved compliance and reduced operational costs.

The Upgrade Guide describes the correct order to perform updates such that all packages continue to perform correctly once upgraded.

The product related release notes and documents are available online at https://docs.delinea.com/.

Delinea software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962. (Ref: CS-44575)

Feature Changes in this Release

For a list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite Release Notes.

For a list of platforms that Delinea will remove support in upcoming releases, refer to the 'Notice of Termination Support' section in the Server Suite Release Notes.

General

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Security Fix

Server Suite DirectControl Agent for

NSS and adquery now supports NTLM name lookup for users from one-way trusted forests.

Enhanced the Domain Controller failover algorithm so that now the default behavior is that the UNIX Agent (adclient) will try to pick up Domain Controllers based on weighted random selection and re-establish LDAP bindings every 12 hours if adclient.binding.refresh.force is true.

DirectControl Command Line Utilities

Delinea cURL is upgraded based on cURL v7.75.0 instead of v7.70.0.

Configuration Parameters

New Parameters

The new parameters are:

- smartcard.pkcs11.module (replaces rhel.smartcard.pkcs11.module)
- smartcard.login.force (replaces rhel.smartcard.login.force)
- dzdo.timestamp_type

Modified Parameters

The following configuration parameters are deprecated (but are still supported for backward compatibility) :

- rhel.smartcard.pkcs11.module
- rhel.smartcard.login.force

Audit Trail Events

Server Suite Access Manager

Server Suite Access Module for PowerShell

Server Suite Group Policy Management

Server Suite Licensing Service

Server Suite OpenLDAP Proxy

Server Suite OpenSSH

DirectControl Openssh was upgraded based on openssh-8.8p1.

Server Suite OpenSSL

OpenSSL was upgraded from 3.0.0 to 3.0.1 with the CVE-2022-0778 patch applied.

Server Suite Report Services

You can now set gMSA as a service account through Report Service Silent Configuration.

Server Suite Smart Card

Added Smart Card Support for Ubuntu 21.10.

Server Suite Windows Installer

Server Suite Windows SDK

Server Suite Zone Provisioning Agent

Fixed Issues in this Release

General

Fix an issue where newly provisioned zone users were not listed as their parent zone groups' members. This issue occurred with AD users that were already added to the corresponding AD groups but were newly provisioned in the zone.

Fixed an issue where the user could not do an MFA log in with RADIUS MFA authentication.

Fixed an issue where the pass-through feature didn't work when a user tried to log in using MFA ssh.

Security Fix

Server Suite DirectControl Agent for

Fixed adclient so that it respects what is set in the "adclient.group.ignore.blocked.domain.member" configuration parameter so that adclient ignores groups from intentionally blocked domains when checking computer role groups.

Fixed an issue where the UNIX Agent (adclient) or CLIs produced the "No credentials found with supported encryption types" Kerberos error message after some customers changed the "adclient.krb5.permitted.encryption.types" and "adclient.krb5.tkt.encryption.types" settings.

Fixed an issue where, in some situations, the kset.altupn and krb5.conf settings were not updated as expected.

Fixed an issue where the NSS user lookup fails sometimes to find an zone user when the configuration "nss.user.group.prefer.cache: true" is in place and the corresponding AD user account is from another forest.

Fixed a library conflict between OpenSSL and DB2 server on AIX. This fixed a problem with using the DirectControl DB2 GSSAPI Kerberos plugin (Single Sign On) on AIX.

DirectControl Command Line Utilities

Fixed an issue where dzdo could not login using single sign-on to a remote host to run a command.

Fixed some issues in the CAPI cache, which is used by the DirectControl add-on packages (such as the adbindproxy and DB2 plugins), smart card support and macOS. Note that the CAPI cache is disabled by default prior to DirectControl version 5.8.1.

Fixed an issue where ssh SSO login failed when working with gssproxy.

DirectControl Installation

Fixed a problem where some services may not be registered after installing the packages on Solaris 10.

Audit Trail Events

Server Suite Access Manager

Server Suite Access Module for PowerShell

Server Suite Group Policy Management

Server Suite Licensing Service

Server Suite NIS

Fixed an issue with the NIS daemon loading snapshot files. You can now make the NIS daemon load all the snapshot files into memory. Doing this can be more efficient to respond ypGetAll, but it will also have a bigger memory footprint. Please contact support if you want to use this feature.

Fixed an issue where adnisd sometimes would get incomplete group NIS maps.

Server Suite OpenLDAP Proxy

Server Suite OpenSSH

Enhanced the GSSAPIKexAlgorithms to support more algorithms. Now DirectControl Openssh supports the following 7 algorithms:

- "gss-group1-sha1-"
- "gss-group14-sha1-"
- "gss-group14-sha256-"
- "gss-group16-sha512-"
- "gss-gex-sha1-"
- "gss-nistp256-sha256-"
- "gss-curve25519-sha256-"

Fixed an issue where the DirectControl Openssh systemd service file is not installed on SuSE15 SP3.

Fixed an issue where the DirectControl Openssh match block set in group policies were lost in sshd_config.

Server Suite Report Services

Server Suite Smart Card

Fixed an issue with Citrix VDA smartcard authentication on Debian or Ubuntu systems where some customers are using 3rd-party pkcs11 libraries.

Server Suite Windows Installer

Server Suite Windows SDK

Server Suite Zone Provisioning Agent

Fixes in Release 2022 Component Update

Fixed several critical security fixes by upgrading 'Centrify OpenSSL' to 3.0.7. (Ref: 469890)

Fixed issues related to CVE-2022-42915 by upgrading cURL to 7.86.0. (Ref: 469891)

Fixed an issue where if you were using Release 2021 (5.8.0) or earlier and you upgraded to the Release 2022 (5.9.0) GA, you might have encountered a cache upgrade issue. The issue caused login failures for offline upgrades and had a negative impact on *NIX agent performance for large Active Directory environments. If you're planning to upgrade, we highly recommend that you use this updated version. You know you have the updated version when you run `adinfo -v` and you get CentrifyDC 5.9.0-159.

Applied the patches of CVE-2022-1292, CVE-2022-1473, CVE-2022-1434 and CVE-2022-1343 to Centrify OpenSSL.

If the domain controller has installed Windows updates dated November 9, 2021 or later and set the new registry value "PacRequestorEnforcement" as "2", resetting passwords by way of Kerberos would fail. As a result, DirectControl adjoin, adkeytab and adpasswd command line utilities would fail to reset the accounts' password. Microsoft has confirmed this issue, please see the Known issues section of this article. This release has a solution to bypass that issue.

Known Issues

The following sections describe common limitations or known issues associated with this Authentication Service and Privilege Elevation Service release.

For the most up to date list of known issues, please login to the Customer Support Portal at https://www.delinea.com/support and refer to Knowledge Base articles for any known issues with the release.

Server Suite DirectControl Agent for

- Known Issues with Multi-Factor Authentication (MFA)

  If MFA is enabled but the parameter "adclient.legacyzone.mfa.required.groups" is set to a non-existent group, all AD users will be required for MFA. The workaround is to remove any non-existent groups from the parameter. (Ref: CS-39591b)

- Known Issues with AIX

  On AIX, upgrading DirectControl agent from 5.0.2 or older versions in disconnected mode may cause unexpected behavior. The centrifydc service may be down after upgrade. It's recommended not to upgrade DirectControl agent in disconnected mode. (Ref: CS-30494a)

  Some versions of AIX cannot handle username longer than eight characters. As a preventive measure, we have added a new test case in the adcheck command to check if the parameter LOGIN_NAME_MAX is set to 9. If yes, adcheck will show a warning so that users can be aware of it. (Ref: CS-30789a)

- Known issues with Fedora 19 and above (Ref: CS-31549a, CS-31730a)

  There are several potential issues on Fedora 19 and above:

  1. The adcheck command will fail if the machine does not have Perl installed.
  2. Group Policy will not be fully functional unless Text/ParseWords.pm is installed.

- Known issues with RedHat

  When logging into a RedHat system using an Active Directory user that has the same name as a local user, the system will not warn the user of the conflict, which will result in unpredictable login behavior. The workaround is to remove the conflict or login with a different AD user. (Ref: CS-28940a, CS-28941a)

  - Known issues with rsh / rlogin (Ref: IN-90001)
  - When using rsh or rlogin to access a computer that has DirectControl agent installed, and where the user is required to change their password, users are prompted to change their password twice. Users may use the same password each time they are prompted, and the password is successfully changed.

- Known issues with compatibility

  Using DirectControl 4.x agents with Access Manager 5.x (Ref: IN-90001)

- DirectControl 4.x agents can join classic zones created by Access Manager 5.x. It will ostensibly be able to join a DirectControl 4.x agent to a hierarchical zone as well, but this causes failure later as such behavior is undefined.

Default zone not used in DirectControl 5.x (Ref: IN-90001)

- In DirectControl 4.x, and earlier, there was a concept of the default zone. When Access Manager was installed, a special zone could be created as the default zone. If no zone was specified when joining a domain with adjoin, the default zone would be used.
- This concept has been removed from DirectControl 5.0.0 and later as it is no longer relevant with hierarchical zones. In zoned mode, a zone must now always be specified.
- A zone called "default" may be created, and default zones created in earlier versions of Access Manager may be used, but the name must be explicitly used.

Smart Card

- Release 18.8 includes an update to Coolkey to support Giesecke & Devrient 144k, Gemalto DLGX4-A 144, and HID Crescendo 144K FIPS cards. However, this has caused known issues that may cause CAC cards to only work sporadically. A workaround for CAC cards is to wait for it to prompt for PIN and Welcome, without removing the card, and then try again. (Ref: CC-58013a)
- There is a Red Hat Linux desktop selection issue found in RHEL 7 with smart card login. When login with smart card, if both GNOME and KDE desktops are installed, user can only log into GNOME desktop even though "KDE Plasma Workspace" option is selected. (Ref: CS-35125a)
- On RHEL 5.10 and 5.11, if "Smart Card Support" is enabled and a smartcard is inserted on the login screen, a PIN prompt may not show up until you hit the "Enter" key. The workaround is to replace libsoftokn3.so with the old one on RHEL 5.9, which is a shared object file in NSS package. (Ref: CS-35038a)
- On RHEL 5.10 and 5.11, if "Smart Card Support" is enabled and "Card Removal Action" is configured as "Lock", the screen will be locked several seconds after login with smart card. The workaround is to replace libsoftokn3.so with the old one on RHEL 5.9, which is a shared object file in NSS package. (Ref: CS-33871a)
- When a SmartCard user attempts to login on Red Hat 6.0 with a password that has expired, the authentication error message may not mention that authentication has failed due to an expired password. (Ref: CS-28305a)
- On RedHat, any SmartCard user will get a PIN prompt even if he's not zoned, even though the login attempt will ultimately fail. This is a divergence from Mac behavior - On Mac, if a SmartCard user is not zoned, Mac doesn't even prompt the user for PIN. (Ref: CS-33175c)
- If a SmartCard user's Active Directory password expires while in disconnected mode, the user may still be able to log into their machine using their expired password. This is not a usual case, as secure SmartCard AD environments usually do not allow both PIN and Password logins while using a Smart Card. (Ref: CS-28926a)
- To login successfully in disconnected mode (Ref: CS-29111a):
  - For a password user:
    - A password user must log in successfully once in connected mode prior to logging in using disconnected mode. (This is consistent with other DirectControl agent for *NIX behavior)
  - For a SmartCard user:
    - The above is not true of SmartCard login. Given a properly configured RedHat system with valid certificate trust chain and CRL set up, a SmartCard user may successfully login using disconnected mode even without prior successful logins in connected mode.
    - If certificate trust chain is not configured properly on the RedHat system, the SmartCard user's login attempt will fail.
    - If the SmartCard user's login certificate has been revoked, and the RedHat system has a valid CRL that includes this certificate, then the system will reject the user.
- After upgrading from DirectControl version 5.0.4 to version 5.1, a Smartcard user may not be able to login successfully. The workaround is to run the following CLI commands:

  sudo rm /etc/pam_pkcs11/cacerts/*

  sudo rm /etc/pam_pkcs11/crls/*

  sudo rm /var/centrify/net/certs/*

  then run adgpupdate. (Ref: CS-30025c)
- When CRL check is set via Group Policy and attempting to authenticate via Smartcard, authentication may fail. The workaround is to wait until the

Group Policy Update interval has occurred and try again or to force an immediate Group Policy update by running the CLI command adgpupdate. (Ref: CS-30090c)

- After upgrading from DirectControl agent Version 5.0.4 to version 5.1.1, a SmartCard user may not be able to authenticate successfully. The workaround is to perform the following CLI command sequence:

```
sctool -d
sctool -e
sudo rm /etc/pam_pkcs11/cacerts/*
sudo rm /etc/pam_pkcs11/crls/*
sudo rm /var/centrify/net/certs/*"
adgpupdate
```

and then re-login using the SmartCard and PIN. (Ref: CS-30353c)

- A name-mapping user can unlock screen with password even though the previous login was with PIN. (Ref: CS-31364b)
- Need to input PIN twice to login using CAC card with PIN on RedHat. It will fail on the first input but succeed on the second one. (Ref: CS-30551c)
- Running "sctool –D" with normal user will provide wrong CRL check result. The work-around is to run it as root. (Ref: CS-31357b)
- Screen saver shows password not PIN prompt (Ref: CS-31559a)

Most smart card users can log on with a smart card and PIN only and cannot authenticate with a username and password. However, it is possible to configure users for both smart card/PIN and username/password authentication. Generally, this set up works seamlessly: the user either enters a username and password at the log on prompt, or inserts a smart card and enters a PIN at the prompt.

However, for multi-user cards, it can be problematic when the screen locks and the card is in the reader. When a user attempts to unlock the screen, the system prompts for a password, not for a PIN, although the PIN is required because the card is in the reader. If the user is not aware that the card is still in the reader and enters his password multiple times, the card will lock once the limit for incorrect entries is reached.

On RHEL 7, an authenticated Active Directory user via smart card cannot login again if the smart card is removed. This is due to a bug in RHEL 7, https://bugzilla.redhat.com/show_bug.cgi?id=1238342. This problem does not happen on RHEL6. (Ref: CSSSUP-6914c)

Report Services

- N/A

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

- 2022.1 Windows Agent Release Notes
- 2022 Windows Agent Release Notes

For the list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite release notes.

For the platforms to be removed support in coming releases, refer to the "Notice of Termination Support" section in the Server Suite release notes.

About Server Suite Agent for Windows

The Server Suite Agent for Windows package contains software to support auditing, access control, and privilege management on Windows computers. Audit and Access features must be installed together but their services can be enabled separately on the Windows computers you want to manage.

For auditing, the Server Suite Agent for Windows requires the Auditing & Monitoring Service feature set, which is available in Server Suite. Auditing & Monitoring Service enables detailed auditing of user activity on a wide range of UNIX, Linux and Windows computers. With Auditing & Monitoring Service, you can perform immediate, in-depth troubleshooting by replaying user activity that may have contributed to system failures, spot suspicious activity by monitoring current user sessions, and improve regulatory compliance and accountability by capturing and storing detailed information about the applications used and the commands executed. If you enable auditing, the Server Suite Agent for Windows records user activity on the Windows computer when it is installed.

For access control and privilege management, the Server Suite Agent for Windows requires the Authentication Service and Privilege Elevation Service feature sets, which are available in Server Suite. With Authentication Service and Privilege Elevation Service, you can configure and manage role-based access controls for Windows servers. The Server Suite Agent for Windows extends the access control and privilege management features available for Linux and UNIX computers, so that you can use a single console to manage multiple platforms. You can deploy the Server Suite Agent for Windows in a Windows-only environment or as part of a mixed environment that includes Windows, Linux, and UNIX computers.

The Server Suite Agent for Windows provides both privilege elevation and auditing functionalities, and for more information about the auditing feature, refer to the Delinea Auditing & Monitoring Service Release Notes for more detailed information.

You can obtain information about previous releases from the Delinea Support Portal, in the Documentation & Application Notes page.

Delinea software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962. (Ref: CS-44575)

Feature Changes

Security Fixes

General Changes

Fixed Issues

We have fixed the memory allocation issues related to Microsoft KB: KB5014697 (Win 11) / KB5014692 (Win10) / KB5014699 (Win2019) / KB5014702 (Win2016) KB updates.

Known Issues

Installation and Uninstallation

- Upgrading from the beta build to this version may result in offline MFA mode if there are multiple authentication servers registered in your AD forest. To resolve this, uninstall the beta build first and then install this new version. (Ref: CS-41915)
- Upgrading Windows while the Agent is installed will result in a failure of the Windows upgrade. The workaround is to uninstall the agent before performing the upgrade or perform a fresh Windows install without keeping existing applications and settings. (Ref: CS-42200)
- Currently the MFA login feature is not supported on Windows Server 2016 "Server Core" systems. This feature component will not be installed on Windows Server 2016 "Server Core" systems. (Ref: CS-42192, CS-42527)
- The Delinea Common Component should be the last Delinea Server Suite component uninstalled. If the component is uninstalled before other component, it must be reinstalled by the uninstall process to complete its task. (Ref: 36226a)
- If you intend to install the software on the desktop with elevated privilege, you should not check the “Run with UAC restrictions” option when creating the desktop. (Ref: 39725b)
- When you double-click on the Server Suite Agent for Windows msi and select the “repair” option, the existing files are replaced irrespective of their

version number, even when they are identical. As a result, a prompt to restart the system is displayed as files that were in use were replaced. However, if you use the Easy Installer to do the repair and a file on the disk has the same version as the file that is part of the installer package, the installed file will not be replaced. Therefore, there will not be any prompt to restart the system. (Ref: 26561a)

- When the Server Suite Agent for Windows is either installed or uninstalled and the prompt for a machine restart is deferred using the "restart later" option or ignored, other components of DirectManage may display errors due to an incomplete installation. A restart is mandatory if requested after install or uninstall operation. (Ref: 36307a)
- Users may notice a few "Side by side" configuration errors in the Event Viewer after installing the Server Suite Agent for Windows, if Microsoft KB945140 related components have been installed on the local machine previously. These errors will go away after you restart the computer and have no functional effect. (Ref: 35302a)
- If you uninstall the Server Suite Agent for Windows while the DirectAudit Agent Control Panel is open, files needed by the uninstall process may be blocked. You should close the DirectAudit Agent Control Panel for a successful conclusion to the uninstall process. (Ref: 25753a)
- If you have installed the Access feature of Server Suite Agent for Windows from Server Suite 2013 and are trying to upgrade the component to the latest version, you may see the following error during the installation process, "Service 'DirectAuthorize Agent' could not be installed. Verify that you have sufficient privileges to install system services." If you see this error message, it typically indicates that the existing service is taking longer time to stop and hence the new service is not getting installed. When you see this error, wait for some time (typically 30 seconds) and click on Retry button on the error message box. (Ref: 47270a)
- Server Suite Agent for Windows and its installer are built on .NET. Therefore, .NET is always installed as a pre-requisite before the agent is installed. If .NET is removed from the system later, Server Suite Agent for Windows will not run properly. User will also experience problem when trying to remove Server Suite Agent for Windows from the system. To properly uninstall Server Suite Agent for Windows, please make sure Server Suite Agent for Windows is uninstalled before .NET. (Ref: 39051a)
- The list of rescue users is stored in different places in Suite 2013.3 (or previous releases) and Suite 2014 and this list is not automatically migrated to its new location when upgrading from Suite 2013.3 or a previous release to Suite 2014. Because of this, it's highly recommended that Server Suite Agent for Windows should not be upgraded in disconnected mode (i.e. when the system cannot connect to the Active Directory). If a system is upgraded in disconnected mode, the list of rescue users will be lost and only local administrators will be able to login to the system after reboot. (Ref: 57622a)
- If you install Access feature of Server Suite Agent for Windows without installing the Audit feature, the registry key value for HKEY_LOCAL_MACHINE\SOFTWARE\Centrify\AuditTrail\AuditTrailTargets is set to zero as expected, which means the audit trail is not sent to DirectAudit Audit Store database. However, if you try to change the installed features list of Server Suite Agent for Windows and add the Audit feature later, the change process does not automatically set the AuditTrailTargets value to the expected new value of 1, which means to send audit trail data to DirectAudit Audit Store database. This is a known issue and workaround is to set this value manually to 1 after the installer finishes the process of adding new feature. (Ref: 59353b)
- If you have installed the Access feature of Server Suite Agent for Windows from earlier version and then upgraded the component to the latest version while the Agent for Windows is not currently connected to any Active Directory domain controller, only users who have been assigned a role with rescue rights will be able to log on to the computer until the connection to Active Directory is restored. (Ref: 58858b)

Configuration

- In a cross-forest environment, forest A user cannot enroll a device joined to forest B when forest A does not have a connector. (Ref: CS-44805)
- When a machine that has MFA for Windows login enabled is reconfigured to connect to a different forest, the previous setting for the authentication server may no longer be valid. However, if there are Group Policy login settings applied to the machine in the new forest, the new settings will be enabled when the Group Policy Editor refreshes. (Ref: CS-41928)
- In Windows 2016 and Win10, during the login process, selecting SMS or using other mechanisms like Security Question/Phone call/Password/Email/Mobile for MFA and clicking the "Commit" button will be intermittently unresponsive. (Ref: CS-41699)
- It can take a long time for users in offline mode to be re-prompted for their passcode.

  In the event that the Agent cannot connect to the Delinea Authentication Server, and a user is required to enter an offline passcode, it can take up to several minutes for the agent to re-prompt for the passcode if the user enters it incorrectly. If the user tries to cancel login by pressing "back" or by switching the user, the login screen may become unresponsive. This time lag between an incorrect passcode and the re-prompt can also occur when a user incorrectly enters their offline passcode for privilege elevation. (Ref: CS-41302)
- Administrator should always leave the zone before joining the computer to a different domain. Otherwise, DirectAuthorize may not function correctly after the computer is joined to a different domain. (Ref: 54278b)

- In some large environment with multiple domain controllers, it may take up to one minute for the new zone setting in Server Suite Agent Configuration to take effect. (Ref: 58128b)
- If one of the Global Catalog servers is unavailable, user may not be able to configure the zone for Server Suite Agent for Windows. (Ref: 58621b)
- Microsoft normally automatically distributes and installs root certificates to the Windows system from trusted Certificate Authorities (CA) and users are seamlessly able to use a secure connection by trusting a certificate chain issued from the trusted CA. However, this mechanism may fail if the system is in a disconnected environment where access to Windows Update is blocked or this feature of automatic root certificate installation is disabled. Without updates on the certificate trust list (CTL), the default CTLs on the system may not be adequate for secure connections of Delinea multi-factor authentication especially for older versions of Windows. To ensure the success of Delinea multi-factor authentication, user may need manually distribute and install the latest CTLs and the required root certificate to systems in a disconnected environment. See Delinea KB-6724 for further information. (Ref: CS-39703)

Environment

- On Windows 10 and Win2K16 machines with Delinea Privilege Elevation Service, following will occur (Ref: CS-43883):

• Pop up an error dialog several seconds after clicking "Open file location" in the context menu of a shortcut on the start menu. Explorer windows will display correctly.

• No responses to the following actions

• Clicking "Open file location" in the context menu of a shortcut on desktop

• Clicking "Open file location" in the context menu of a shortcut on the Delinea Start menu in the Privileged Desktop

• Slow response to "OK", "Cancel" in the shortcut property page after "Open file location" in the general tab is clicked. The dialog will close after several seconds.

- On some Windows 10 machines, the smart card login option may not be displayed if another credential method has been recently used. To display the smart card login option, remove and insert a smart card into the reader. This will cause the login screen to reload and will display the smart card login option. (Ref: CS-41282)
- Server Suite Agent for Windows requires you to patch to at least build 10.0.14393 on Windows 10 and Windows Server 2016 to use MFA features. (Ref: CS-41387)
- Selective two-way external trusts are not supported. Both Windows machines and Server Suite zones are required to be in the same forest or different forests with a two-way forest trust established. (Ref: 40713b, 44644b, 44647b, 44657b, 40643b, 40650b, 45341b, 45372b)
- Environment with no Global Catalog is not supported. (Ref: 46577a)
- DirectAuthorize for Windows requires machine time to be synchronized with domain controller. VMware virtual machine has a known issue that its time may not be synchronized with domain controller. This problem occurs more often on an overloaded virtual machine host. If the system clocks on the local Windows computer and the domain controller are not synchronized, DirectAuthorize for Windows does not allow any domain users to login. You can try the following KB from VMware to fix the time synchronization issue. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1189 (Ref: 47795b)

RunAsRole

- If you use the “RunAsRole.exe /wait” command to run a Python script, the input/output cannot be redirected for versions of Python below 3.0.0. (Ref: 45061a)
- Run As Role menu is not available on the start screen in Windows 8 or Windows 2012 or later because Microsoft doesn’t support any custom context menu on the start screen. User has to go to the Windows desktop in order to launch an application using Run As Role context menu. (Ref: 35487a)
- On Windows 8, Windows 8.1, Windows Server 2012 and Windows Server 2012 R2, use “Run as role” with Local Administrators group privilege on Control Panel does not have sufficient permission to add printers if the printer drivers are not pre-installed on the computer. The workaround is to define a role run as a user with Local Administrator privilege or with a group as a member of Local Administrators group. (Ref: 68826a)
- The “Run as role” for Windows Media Player is not recommended. Please use privilege desktop instead. (Ref: 55615a)
- When running “RunAsRole.exe /wait sc.exe” with no argument provided to sc.exe, sc.exe will prompt

Would you like to see help for the QUERY and QUERYEX commands? [y I n]:

Typing 'y' or 'n' doesn't do anything because the input cannot be successfully redirected to sc.exe. (Ref: 47016b)

- It is not recommended to change zone via Run As Role since the role that is in use may no longer be available once after leaving from the previous zone during the change zone process. (Ref: 58043a)

Desktop with Elevated Privileges

- In desktop with elevated privilege a mouse left click does not work for SysTray Icons that involve opening the WinRT(or new Window) UI. The Systray icons affect are Time, Volume Control, or any third party icons on Windows 10/2016. (Ref: CS-39454)
- Server Manager cannot be started on multiple desktops at the same time. The bug exists on Windows 2012R2, 2016. (Ref: CS-42060)
- On a desktop with elevated privileges on Windows 8, 10 & 2016, the search for files or folders will be intermittently disabled from the Start menu ("Start" menu > "Search" > "For Files or Folders..."). (Ref: CS-42066)
- On a desktop with elevated privileges, if you open the Task Manager and select "*File > New* Task" to run an application without selecting the "Create this task with administrative privileges" option, the application will be launched on the default desktop. This issue occurs when User Account Control (UAC) is enabled. (Ref: 32169a)
- If the sAMAccountName attribute of an Active Directory account is changed while the old account name is still cached on the computer, you may see the following error message when creating a new desktop or using "Run as role" with a right configured to run as the modified user account:

  "Failed to open new desktop. Right xxx references bad user account."

  The workaround is to restart the computer. (Ref: 35124a)
- On a desktop with elevated privileges, if you use "Control Panel > Programs > Programs and Features" to uninstall a program, you may see the following warning message and cannot uninstall the software.

  "The system administrator has set policies to prevent this installation."

  This issue happens when User Account Control (UAC) is enabled and when "Run with UAC restrictions" is selected when creating the new desktop. (Ref: 33384a)
- When you open the Start menu "Help and Support" item on a desktop with elevated privileges, the Windows Help and Support is launched on the default desktop. Switch to the default desktop to view the information. (Ref: 32147a)
- If you shut down, restart, or log off from a desktop with elevated privileges, all running applications are terminated forcibly without being prompted to save any open documents. (Ref: 40749a)
- You cannot launch Windows Security Options using "Start menu -> Windows Security" on a privilege desktop with elevated privileges when using a remote desktop connection. You must switch back to the default desktop to continue. (Ref: 45995b)
- Installation of IE9 on desktops with elevated privileges may cause the privileged desktop to become unusable. Use "RunAsRole" for installation of IE9 instead. (Ref: 44930a)
- You cannot use the Start menu option "Switch User" while you are using a role-based, privileged desktop. To use the "Switch User" shortcut, change from the privileged desktop to your default Windows desktop. From the default desktop, you can then select Start > Switch User to log on as a different user. (Ref: 39011b)
- On a DirectAuthorize desktop using a role with local administrator privilege, the Stand By option in the shutdown menu does not work. This is a known issue and will be addressed in future release. (Ref: 58280a)
- VMWare registers to run VMwareUser.exe on the guest operating system to enable user to copy and paste text between the guest and managed host operating systems. Creating multiple desktops with different user accounts causes multiple VMwareUser.exe are run in different user accounts in the same logon session. VMwareUSer.exe cannot support this scenario and therefore an error message is displayed on the default desktop which blocks all the UI operation on the new desktop. To workaround this problem, user can disable the VMWare user program on the guest machine by deleting the registry value name "VMware User Process" from HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. (Ref: 49268a)
- On a privileged desktop on Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2, you may not be able to access the VMware shared folder. (Ref: 40686c)

- Windows logo key keyboard shortcuts are not supported on privileged desktop. Depends on the key and operation system, the shortcut could either have no effect or its effect is applied to the default desktop instead. (Ref: 47588b)
- A Start menu on privileged desktop on Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2, to make up for a limitation of Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2. In addition, navigating to a Modern Start screen to use Modern-style apps from a privileged desktop is not possible from either the charm bar or using a Windows. Note: you must switch back to the default desktop in order to go to Modern Start screen. (Ref: 41245c)
- The Challenge Pass-Through Duration setting is currently not supported for Server Suite Windows multi-factor authentication. The Challenge Pass-Through Duration setting does not require a user, who has successfully met a multi-factor authentication challenge, to re-authenticate through mfa to use an mfa-required right or role if that user chooses the same challenge mechanism when prompted within the duration specified in the setting. (Ref: CS-39432)

Roles and Rights

- No 'Require multi-factor authentication' system right for the predefined 'Windows Login' role. To define this system right for MFA, use the pre-defined Require MFA for logon role, or create a new custom role. (Ref: CS-40888)
- Windows Network Access rights do not take effect on a Linux or UNIX machines. If you select a role to start a program or create a desktop that contains a Network Access right, you can only use that role to access Windows computers. The Windows computers you access over the network must be joined to a zone that honors the selected role. The selected role cannot be used to access any Linux or UNIX server computers on the network. (Ref: 32980a)
- To elevate privileges to the “Run as” account specified in a Windows right, the “run as” account must have local logon rights. If you have explicitly disallowed this right, you may receive an error such as “the user has not been granted the requested logon type at this computer” when attempting to use the right. (Ref: 34266a)
- If your computer network is spread out geographically, there may be failures in NETBIOS name translation. If a NETBIOS name is used, Active Directory attempts to resolve the NETBIOS name based on the domain controller that the user belongs to, which in a multi-segment network might fail. Therefore, Network Access rights might not work as expected if the remote server is located using NETBIOS name. You may need to consult your network administrator to work around this issue. (Ref: 39087a)
- File hash matching criteria in the Application right is not supported for a file larger than 500MB. This is to make sure DirectAuthorize does not spend too much CPU and memory resources to calculate the file hash. User trying to import a file with the size larger than 500MB will see an empty value for the file hash field. (Ref: 56778a)
- For a small set of applications, enabled matching criterion: “Product Name”, “Product version”, “Company”, “File Version” or “File Description” of a Windows Application Right may fail to match after upgrading agent under the following conditions: Any value for the enabled matching criteria is defined by either import from a process or file * The matching criteria is defined by 5.1.3 or 5.2.0 DirectManage Access Manager since the number of affected application is expected to be relatively low, proactively updating the defined matching criteria of Windows Application Right is not necessary. (Ref: 60053a)
- Smart card users can continue to use their own smart card to logon, but there will be no Server Suite MFA features applied to smart card users. (Ref: CS-41539)

Compatibility with 3rd Party Products

- MFA may be skipped when connecting through XenDesktop because the Citrix Credential Provider may be used instead of the Delinea Credential Provider. The workaround is to disable the Citrix Credential Provider through the Group Policy "Centrify Settings\Windows Settings\MFA Settings\Specify the credential providers to exclude from the logon screen." (Ref: CS-46744)
- VirtualDesktop is not compatible with Server Suite Agent for Windows. Users should use the Delinea system tray applet to create virtual desktop instead. (Ref: 44641b)
- Attempting to launch SCOM Operation Console on privileged desktops will fail if there is an existing instance on other desktops and a new SCOM Operation Console will be started on the desktop with the existing instance. The workaround is to close all existing instances before starting a new SCOM Operation Console. (Ref: CS-43790)
- The startup path for “SharePoint 2010 Management Shell” and “Exchange Management Shell” may set to C:\Windows instead of user home directory if it is launched via RunAsRole.exe or from a desktop with elevated privilege. (Ref: 38814b, 46943b)
- On a desktop with elevated privileges, if you install McAfee Security Scan products and click “View Readme”, the Readme.html is shown on the default desktop. Similar issues may happen with other third party programs. The alternate way to view the Readme.html on the desktop of a managed computer

is to open the Readme.html file directly. (Ref: 34642a)

- Attempting to enable Kerberos authentication for Oracle databases will fail. This issue is being brought to the attention of Oracle Support for a resolution in upcoming releases. (Ref: 33835b)
- The Microsoft Snipping Tool utility has a bug that prevents it from running on a desktop with elevated privileges. (Ref: 31931a)
- Some applications do not use the process token to check the group membership. They check the user's group membership on its own. Therefore, any Windows rights configured to use a privileged group will not take effect in these applications. The workaround is to use a privileged user account instead of a privileged group. Here is the list of known application with this issue:

  1. vCenter Server 5.1
  2. SQL Server
  3. Exchange 2010 or above
  4. SCOM 2007

  (Ref: 45318a, 45218a, 43779a, 38016a)
- Privilege elevation using Windows Rights for Internet Explorer (IE) 7 is not supported. (Ref: 33425a)
- Privilege elevation using Windows rights for "Remote Desktop" is not supported. (Ref: 45222b)
- Privilege elevation using Windows rights for taskmgr.exe, explorer.exe, and cmd.exe are not recommended. A user granted privileges with Windows rights is implicitly granted to run any executable under the same privilege. (Ref: 45861a, 40525a)
- Users may notice an error and cannot install ActivClient after installing Server Suite Agent for Windows. During the installation of ActivClient, it attempts to change the local security setting. However, there is a known issue for Server Suite Agent for Windows of blocking the local security setting (Ref: 63609b). Therefore, users may not be able to install ActivClient successfully after installing Server Suite Agent for Windows. We suggest installing ActivClient before installing Server Suite Agent for Windows. If Server Suite Agent for Windows has been installed, please uninstall it and follow the installation sequence suggested. This issue happens on Windows 8.1 and Windows 2012 R2 only. (Ref: 76016b)
- McAfee Virus scan may block the Server Suite Agent for Windows from registering the LSA packages while joining the system to a zone. The zone join operation now detects the same and shows a warning to the user. (Ref: CS-48893)

Application Manager

- Application Manager does not support Server Core edition of Windows. (Ref: CS-40656)
- Application Manager may not be able to generate Audit Trail event for the uninstall, change or repair operations which require reboot. (Ref: CS-45641)

Network Manager

- Network Manager does not support Server Core edition of Windows. (Ref: CS-42675)

Endpoint Enrollment

- When a Windows machine is enrolled by a user as a personal device and subsequently that user is disabled, after upgrading the product, there is no way to let another user to enroll a personal device for that machine. The workaround is to remove the service "Identity Services Platform" in Agent Configuration and add that service again. (Ref: CS-44514)

Server Suite Agent for Windows

- Auditing status is incorrectly displayed on Authorization Center and the desktop notification message when the following Group Policies are enabled:
  - Audited user list
  - Non-audited user list

  (Ref: CS-46321)
- Server Suite Agent for Windows installation may prematurely end on systems that have Citrix Virtual Delivery Agent version 7.9 or higher installed. Please refer to the Delinea Knowledge Base for possible workarounds to deploy Server Suite Agent for Windows on systems affected by this issue. (Ref: CS-46288)

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

For the list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite release notes.

For the platforms to be removed support in coming releases, refer to the "Notice of Termination Support" section in the Server Suite release notes.

About Server Suite Agent for Windows

The Server Suite Agent for Windows package contains software to support auditing, access control, and privilege management on Windows computers. Audit and Access features must be installed together but their services can be enabled separately on the Windows computers you want to manage.

For auditing, the Server Suite Agent for Windows requires the Auditing & Monitoring Service feature set, which is available in Server Suite. Auditing & Monitoring Service enables detailed auditing of user activity on a wide range of UNIX, Linux and Windows computers. With Auditing & Monitoring Service, you can perform immediate, in-depth troubleshooting by replaying user activity that may have contributed to system failures, spot suspicious activity by monitoring current user sessions, and improve regulatory compliance and accountability by capturing and storing detailed information about the applications used and the commands executed. If you enable auditing, the Server Suite Agent for Windows records user activity on the Windows computer when it is installed.

For access control and privilege management, the Server Suite Agent for Windows requires the Authentication Service and Privilege Elevation Service feature sets, which are available in Server Suite. With Authentication Service and Privilege Elevation Service, you can configure and manage role-based access controls for Windows servers. The Server Suite Agent for Windows extends the access control and privilege management features available for Linux and UNIX computers, so that you can use a single console to manage multiple platforms. You can deploy the Server Suite Agent for Windows in a Windows-only environment or as part of a mixed environment that includes Windows, Linux, and UNIX computers.

The Server Suite Agent for Windows provides both privilege elevation and auditing functionalities, and for more information about the auditing feature, refer to the Delinea Auditing & Monitoring Service Release Notes for more detailed information.

You can obtain information about previous releases from the Delinea Support Portal, in the Documentation & Application Notes page.

Delinea software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962. (Ref: CS-44575)

Feature Changes

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Security Fixes

General Changes

Fixed Issues

Fixed Issues in the 2022 Component Update

We have fixed the memory allocation issues related to Microsoft KB: KB5014697 (Win 11) / KB5014692 (Win10) / KB5014699 (Win2019) / KB5014702 (Win2016) KB updates.

Known Issues

Installation and Uninstallation

- Upgrading from the beta build to this version may result in offline MFA mode if there are multiple authentication servers registered in your AD forest. To resolve this, uninstall the beta build first and then install this new version. (Ref: CS-41915)
- Upgrading Windows while the Agent is installed will result in a failure of the Windows upgrade. The workaround is to uninstall the agent before performing the upgrade or perform a fresh Windows install without keeping existing applications and settings. (Ref: CS-42200)
- Currently the MFA login feature is not supported on Windows Server 2016 "Server Core" systems. This feature component will not be installed on Windows Server 2016 "Server Core" systems. (Ref: CS-42192, CS-42527)

- The Delinea Common Component should be the last Delinea Server Suite component uninstalled. If the component is uninstalled before other component, it must be reinstalled by the uninstall process to complete its task. (Ref: 36226a)
- If you intend to install the software on the desktop with elevated privilege, you should not check the "Run with UAC restrictions" option when creating the desktop. (Ref: 39725b)
- When you double-click on the Server Suite Agent for Windows msi and select the "repair" option, the existing files are replaced irrespective of their version number, even when they are identical. As a result, a prompt to restart the system is displayed as files that were in use were replaced. However, if you use the Easy Installer to do the repair and a file on the disk has the same version as the file that is part of the installer package, the installed file will not be replaced. Therefore, there will not be any prompt to restart the system. (Ref: 26561a)
- When the Server Suite Agent for Windows is either installed or uninstalled and the prompt for a machine restart is deferred using the "restart later" option or ignored, other components of DirectManage may display errors due to an incomplete installation. A restart is mandatory if requested after install or uninstall operation. (Ref: 36307a)
- Users may notice a few "Side by side" configuration errors in the Event Viewer after installing the Server Suite Agent for Windows, if Microsoft KB945140 related components have been installed on the local machine previously. These errors will go away after you restart the computer and have no functional effect. (Ref: 35302a)
- If you uninstall the Server Suite Agent for Windows while the DirectAudit Agent Control Panel is open, files needed by the uninstall process may be blocked. You should close the DirectAudit Agent Control Panel for a successful conclusion to the uninstall process. (Ref: 25753a)
- If you have installed the Access feature of Server Suite Agent for Windows from Server Suite 2013 and are trying to upgrade the component to the latest version, you may see the following error during the installation process, "Service 'DirectAuthorize Agent' could not be installed. Verify that you have sufficient privileges to install system services." If you see this error message, it typically indicates that the existing service is taking longer time to stop and hence the new service is not getting installed. When you see this error, wait for some time (typically 30 seconds) and click on Retry button on the error message box. (Ref: 47270a)
- Server Suite Agent for Windows and its installer are built on .NET. Therefore, .NET is always installed as a pre-requisite before the agent is installed. If .NET is removed from the system later, Server Suite Agent for Windows will not run properly. User will also experience problem when trying to remove Server Suite Agent for Windows from the system. To properly uninstall Server Suite Agent for Windows, please make sure Server Suite Agent for Windows is uninstalled before .NET. (Ref: 39051a)
- The list of rescue users is stored in different places in Suite 2013.3 (or previous releases) and Suite 2014 and this list is not automatically migrated to its new location when upgrading from Suite 2013.3 or a previous release to Suite 2014. Because of this, it's highly recommended that Server Suite Agent for Windows should not be upgraded in disconnected mode (i.e. when the system cannot connect to the Active Directory). If a system is upgraded in disconnected mode, the list of rescue users will be lost and only local administrators will be able to login to the system after reboot. (Ref: 57622a)
- If you install Access feature of Server Suite Agent for Windows without installing the Audit feature, the registry key value for HKEY_LOCAL_MACHINE\SOFTWARE\Centrify\AuditTrail\AuditTrailTargets is set to zero as expected, which means the audit trail is not sent to DirectAudit Audit Store database. However, if you try to change the installed features list of Server Suite Agent for Windows and add the Audit feature later, the change process does not automatically set the AuditTrailTargets value to the expected new value of 1, which means to send audit trail data to DirectAudit Audit Store database. This is a known issue and workaround is to set this value manually to 1 after the installer finishes the process of adding new feature. (Ref: 59353b)
- If you have installed the Access feature of Server Suite Agent for Windows from earlier version and then upgraded the component to the latest version while the Agent for Windows is not currently connected to any Active Directory domain controller, only users who have been assigned a role with rescue rights will be able to log on to the computer until the connection to Active Directory is restored. (Ref: 58858b)

Configuration

- In a cross-forest environment, forest A user cannot enroll a device joined to forest B when forest A does not have a connector. (Ref: CS-44805)
- When a machine that has MFA for Windows login enabled is reconfigured to connect to a different forest, the previous setting for the authentication server may no longer be valid. However, if there are Group Policy login settings applied to the machine in the new forest, the new settings will be enabled when the Group Policy Editor refreshes. (Ref: CS-41928)
- In Windows 2016 and Win10, during the login process, selecting SMS or using other mechanisms like Security Question/Phone call/Password/Email/Mobile for MFA and clicking the "Commit" button will be intermittently unresponsive. (Ref: CS-41699)
- It can take a long time for users in offline mode to be re-prompted for their passcode.

In the event that the Agent cannot connect to the Delinea Authentication Server, and a user is required to enter an offline passcode, it can take up to several minutes for the agent to re-prompt for the passcode if the user enters it incorrectly. If the user tries to cancel login by pressing "back" or by switching the user, the login screen may become unresponsive. This time lag between an incorrect passcode and the re-prompt can also occur when a user incorrectly enters their offline passcode for privilege elevation. (Ref: CS-41302)

- Administrator should always leave the zone before joining the computer to a different domain. Otherwise, DirectAuthorize may not function correctly after the computer is joined to a different domain. (Ref: 54278b)
- In some large environment with multiple domain controllers, it may take up to one minute for the new zone setting in Server Suite Agent Configuration to take effect. (Ref: 58128b)
- If one of the Global Catalog servers is unavailable, user may not be able to configure the zone for Server Suite Agent for Windows. (Ref: 58621b)
- Microsoft normally automatically distributes and installs root certificates to the Windows system from trusted Certificate Authorities (CA) and users are seamlessly able to use a secure connection by trusting a certificate chain issued from the trusted CA. However, this mechanism may fail if the system is in a disconnected environment where access to Windows Update is blocked or this feature of automatic root certificate installation is disabled. Without updates on the certificate trust list (CTL), the default CTLs on the system may not be adequate for secure connections of Delinea multi-factor authentication especially for older versions of Windows. To ensure the success of Delinea multi-factor authentication, user may need manually distribute and install the latest CTLs and the required root certificate to systems in a disconnected environment. See Delinea KB-6724 for further information. (Ref: CS-39703)

Environment

- On Windows 10 and Win2K16 machines with Delinea Privilege Elevation Service, following will occur (Ref: CS-43883):

• Pop up an error dialog several seconds after clicking "Open file location" in the context menu of a shortcut on the start menu. Explorer windows will display correctly.

• No responses to the following actions

• Clicking "Open file location" in the context menu of a shortcut on desktop

• Clicking "Open file location" in the context menu of a shortcut on the Delinea Start menu in the Privileged Desktop

• Slow response to "OK", "Cancel" in the shortcut property page after "Open file location" in the general tab is clicked. The dialog will close after several seconds.

- On some Windows 10 machines, the smart card login option may not be displayed if another credential method has been recently used. To display the smart card login option, remove and insert a smart card into the reader. This will cause the login screen to reload and will display the smart card login option. (Ref: CS-41282)
- Server Suite Agent for Windows requires you to patch to at least build 10.0.14393 on Windows 10 and Windows Server 2016 to use MFA features. (Ref: CS-41387)
- Selective two-way external trusts are not supported. Both Windows machines and Server Suite zones are required to be in the same forest or different forests with a two-way forest trust established. (Ref: 40713b, 44644b, 44647b, 44657b, 40643b, 40650b, 45341b, 45372b)
- Environment with no Global Catalog is not supported. (Ref: 46577a)
- DirectAuthorize for Windows requires machine time to be synchronized with domain controller. VMware virtual machine has a known issue that its time may not be synchronized with domain controller. This problem occurs more often on an overloaded virtual machine host. If the system clocks on the local Windows computer and the domain controller are not synchronized, DirectAuthorize for Windows does not allow any domain users to login. You can try the following KB from VMware to fix the time synchronization issue. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1189 (Ref: 47795b)

RunAsRole

- If you use the “RunAsRole.exe /wait” command to run a Python script, the input/output cannot be redirected for versions of Python below 3.0.0. (Ref: 45061a)
- Run As Role menu is not available on the start screen in Windows 8 or Windows 2012 or later because Microsoft doesn’t support any custom context menu on the start screen. User has to go to the Windows desktop in order to launch an application using Run As Role context menu. (Ref: 35487a)

- On Windows 8, Windows 8.1, Windows Server 2012 and Windows Server 2012 R2, use "Run as role" with Local Administrators group privilege on Control Panel does not have sufficient permission to add printers if the printer drivers are not pre-installed on the computer. The workaround is to define a role run as a user with Local Administrator privilege or with a group as a member of Local Administrators group. (Ref: 68826a)
- The "Run as role" for Windows Media Player is not recommended. Please use privilege desktop instead. (Ref: 55615a)
- When running "RunAsRole.exe /wait sc.exe" with no argument provided to sc.exe, sc.exe will prompt

  Would you like to see help for the QUERY and QUERYEX commands? [y | n]:

  Typing 'y' or 'n' doesn't do anything because the input cannot be successfully redirected to sc.exe. (Ref: 47016b)
- It is not recommended to change zone via Run As Role since the role that is in use may no longer be available once after leaving from the previous zone during the change zone process. (Ref: 58043a)

Desktop with Elevated Privileges

- In desktop with elevated privilege a mouse left click does not work for SysTray Icons that involve opening the WinRT(or new Window) UI. The Systray icons affect are Time, Volume Control, or any third party icons on Windows 10/2016. (Ref: CS-39454)
- Server Manager cannot be started on multiple desktops at the same time. The bug exists on Windows 2012R2, 2016. (Ref: CS-42060)
- On a desktop with elevated privileges on Windows 8, 10 & 2016, the search for files or folders will be intermittently disabled from the Start menu ("Start" menu > "Search" > "For Files or Folders..."). (Ref: CS-42066)
- On a desktop with elevated privileges, if you open the Task Manager and select "*File > New* Task" to run an application without selecting the "Create this task with administrative privileges" option, the application will be launched on the default desktop. This issue occurs when User Account Control (UAC) is enabled. (Ref: 32169a)
- If the sAMAccountName attribute of an Active Directory account is changed while the old account name is still cached on the computer, you may see the following error message when creating a new desktop or using "Run as role" with a right configured to run as the modified user account:

  "Failed to open new desktop. Right xxx references bad user account."

  The workaround is to restart the computer. (Ref: 35124a)
- On a desktop with elevated privileges, if you use "Control Panel > Programs > Programs and Features" to uninstall a program, you may see the following warning message and cannot uninstall the software.

  "The system administrator has set policies to prevent this installation."

  This issue happens when User Account Control (UAC) is enabled and when "Run with UAC restrictions" is selected when creating the new desktop. (Ref: 33384a)
- When you open the Start menu "Help and Support" item on a desktop with elevated privileges, the Windows Help and Support is launched on the default desktop. Switch to the default desktop to view the information. (Ref: 32147a)
- If you shut down, restart, or log off from a desktop with elevated privileges, all running applications are terminated forcibly without being prompted to save any open documents. (Ref: 40749a)
- You cannot launch Windows Security Options using "Start menu -> Windows Security" on a privilege desktop with elevated privileges when using a remote desktop connection. You must switch back to the default desktop to continue. (Ref: 45995b)
- Installation of IE9 on desktops with elevated privileges may cause the privileged desktop to become unusable. Use "RunAsRole" for installation of IE9 instead. (Ref: 44930a)
- You cannot use the Start menu option "Switch User" while you are using a role-based, privileged desktop. To use the "Switch User" shortcut, change from the privileged desktop to your default Windows desktop. From the default desktop, you can then select Start > Switch User to log on as a different user. (Ref: 39011b)
- On a DirectAuthorize desktop using a role with local administrator privilege, the Stand By option in the shutdown menu does not work. This is a known issue and will be addressed in future release. (Ref: 58280a)
- VMWare registers to run VMwareUser.exe on the guest operating system to enable user to copy and paste text between the guest and managed host

operating systems. Creating multiple desktops with different user accounts causes multiple VMwareUser.exe are run in different user accounts in the same logon session. VMwareUSer.exe cannot support this scenario and therefore an error message is displayed on the default desktop which blocks all the UI operation on the new desktop. To workaround this problem, user can disable the VMWare user program on the guest machine by deleting the registry value name "VMware User Process" from HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. (Ref: 49268a)

- On a privileged desktop on Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2, you may not be able to access the VMware shared folder. (Ref: 40686c)
- Windows logo key keyboard shortcuts are not supported on privileged desktop. Depends on the key and operation system, the shortcut could either have no effect or its effect is applied to the default desktop instead. (Ref: 47588b)
- A Start menu on privileged desktop on Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2, to make up for a limitation of Windows 8, Windows 8.1, Windows 2012 and Windows 2012 R2. In addition, navigating to a Modern Start screen to use Modern-style apps from a privileged desktop is not possible from either the charm bar or using a Windows. Note: you must switch back to the default desktop in order to go to Modern Start screen. (Ref: 41245c)
- The Challenge Pass-Through Duration setting is currently not supported for Server Suite Windows multi-factor authentication. The Challenge Pass-Through Duration setting does not require a user, who has successfully met a multi-factor authentication challenge, to re-authenticate through mfa to use an mfa-required right or role if that user chooses the same challenge mechanism when prompted within the duration specified in the setting. (Ref: CS-39432)

Roles and Rights

- No 'Require multi-factor authentication' system right for the predefined 'Windows Login' role. To define this system right for MFA, use the pre-defined Require MFA for logon role, or create a new custom role. (Ref: CS-40888)
- Windows Network Access rights do not take effect on a Linux or UNIX machines. If you select a role to start a program or create a desktop that contains a Network Access right, you can only use that role to access Windows computers. The Windows computers you access over the network must be joined to a zone that honors the selected role. The selected role cannot be used to access any Linux or UNIX server computers on the network. (Ref: 32980a)
- To elevate privileges to the "Run as" account specified in a Windows right, the "run as" account must have local logon rights. If you have explicitly disallowed this right, you may receive an error such as "the user has not been granted the requested logon type at this computer" when attempting to use the right. (Ref: 34266a)
- If your computer network is spread out geographically, there may be failures in NETBIOS name translation. If a NETBIOS name is used, Active Directory attempts to resolve the NETBIOS name based on the domain controller that the user belongs to, which in a multi-segment network might fail. Therefore, Network Access rights might not work as expected if the remote server is located using NETBIOS name. You may need to consult your network administrator to work around this issue. (Ref: 39087a)
- File hash matching criteria in the Application right is not supported for a file larger than 500MB. This is to make sure DirectAuthorize does not spend too much CPU and memory resources to calculate the file hash. User trying to import a file with the size larger than 500MB will see an empty value for the file hash field. (Ref: 56778a)
- For a small set of applications, enabled matching criterion: "Product Name", "Product version", "Company", "File Version" or "File Description" of a Windows Application Right may fail to match after upgrading agent under the following conditions: Any value for the enabled matching criteria is defined by either import from a process or file * The matching criteria is defined by 5.1.3 or 5.2.0 DirectManage Access Manager since the number of affected application is expected to be relatively low, proactively updating the defined matching criteria of Windows Application Right is not necessary. (Ref: 60053a)
- Smart card users can continue to use their own smart card to logon, but there will be no Server Suite MFA features applied to smart card users. (Ref: CS-41539)

Compatibility with 3rd Party Products

- MFA may be skipped when connecting through XenDesktop because the Citrix Credential Provider may be used instead of the Delinea Credential Provider. The workaround is to disable the Citrix Credential Provider through the Group Policy "Centrify Settings\Windows Settings\MFA Settings\Specify the credential providers to exclude from the logon screen." (Ref: CS-46744)
- VirtualDesktop is not compatible with Server Suite Agent for Windows. Users should use the Delinea system tray applet to create virtual desktop instead. (Ref: 44641b)
- Attempting to launch SCOM Operation Console on privileged desktops will fail if there is an existing instance on other desktops and a new SCOM

Operation Console will be started on the desktop with the existing instance. The workaround is to close all existing instances before starting a new SCOM Operation Console. (Ref: CS-43790)

- The startup path for "SharePoint 2010 Management Shell" and "Exchange Management Shell" may set to C:\Windows instead of user home directory if it is launched via RunAsRole.exe or from a desktop with elevated privilege. (Ref: 38814b, 46943b)
- On a desktop with elevated privileges, if you install McAfee Security Scan products and click "View Readme", the Readme.html is shown on the default desktop. Similar issues may happen with other third party programs. The alternate way to view the Readme.html on the desktop of a managed computer is to open the Readme.html file directly. (Ref: 34642a)
- Attempting to enable Kerberos authentication for Oracle databases will fail. This issue is being brought to the attention of Oracle Support for a resolution in upcoming releases. (Ref: 33835b)
- The Microsoft Snipping Tool utility has a bug that prevents it from running on a desktop with elevated privileges. (Ref: 31931a)
- Some applications do not use the process token to check the group membership. They check the user's group membership on its own. Therefore, any Windows rights configured to use a privileged group will not take effect in these applications. The workaround is to use a privileged user account instead of a privileged group. Here is the list of known application with this issue:

  1. vCenter Server 5.1
  2. SQL Server
  3. Exchange 2010 or above
  4. SCOM 2007

  (Ref: 45318a, 45218a, 43779a, 38016a)
- Privilege elevation using Windows Rights for Internet Explorer (IE) 7 is not supported. (Ref: 33425a)
- Privilege elevation using Windows rights for "Remote Desktop" is not supported. (Ref: 45222b)
- Privilege elevation using Windows rights for taskmgr.exe, explorer.exe, and cmd.exe are not recommended. A user granted privileges with Windows rights is implicitly granted to run any executable under the same privilege. (Ref: 45861a, 40525a)
- Users may notice an error and cannot install ActivClient after installing Server Suite Agent for Windows. During the installation of ActivClient, it attempts to change the local security setting. However, there is a known issue for Server Suite Agent for Windows of blocking the local security setting (Ref: 63609b). Therefore, users may not be able to install ActivClient successfully after installing Server Suite Agent for Windows. We suggest installing ActivClient before installing Server Suite Agent for Windows. If Server Suite Agent for Windows has been installed, please uninstall it and follow the installation sequence suggested. This issue happens on Windows 8.1 and Windows 2012 R2 only. (Ref: 76016b)
- McAfee Virus scan may block the Server Suite Agent for Windows from registering the LSA packages while joining the system to a zone. The zone join operation now detects the same and shows a warning to the user. (Ref: CS-48893)

Application Manager

- Application Manager does not support Server Core edition of Windows. (Ref: CS-40656)
- Application Manager may not be able to generate Audit Trail event for the uninstall, change or repair operations which require reboot. (Ref: CS-45641)

Network Manager

- Network Manager does not support Server Core edition of Windows. (Ref: CS-42675)

Endpoint Enrollment

- When a Windows machine is enrolled by a user as a personal device and subsequently that user is disabled, after upgrading the product, there is no way to let another user to enroll a personal device for that machine. The workaround is to remove the service "Identity Services Platform" in Agent Configuration and add that service again. (Ref: CS-44514)

Server Suite Agent for Windows

- Auditing status is incorrectly displayed on Authorization Center and the desktop notification message when the following Group Policies are enabled:
  - Audited user list

- Non-audited user list

(Ref: CS-46321)

- Server Suite Agent for Windows installation may prematurely end on systems that have Citrix Virtual Delivery Agent version 7.9 or higher installed. Please refer to the Delinea Knowledge Base for possible workarounds to deploy Server Suite Agent for Windows on systems affected by this issue. (Ref: CS-46288)

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

- 2022.1 Auditing Release Notes
- 2022 Auditing Release Notes

About Server Suite Auditing & Monitoring Service

Delinea Server Suite is a product category that includes the following product offerings:

- Privileged Access Service
- Authentication Service
- Privilege Elevation Service
- Auditing & Monitoring Service

The DirectControl Agent provides services for the Authentication Service and Privilege Elevation Service contained in the CentrifyDC packages. The DirectAudit Agent provides services for Auditing & Monitoring Service contained in the CentrifyDA packages.

The Auditing & Monitoring Service is a key component of Server Suite. It enables detailed auditing of user activity on a wide range of UNIX, Linux, and Windows computers. With this service, you can perform immediate, in-depth troubleshooting by replaying user activity that may have contributed to system failures, spot suspicious activity by monitoring current user sessions, improve regulatory compliance, and ensure accountability by capturing and storing detailed information about the applications used and the commands executed. If you enable auditing, the Server Suite Agent for Windows records user activity on the Windows computer when it is installed. Auditing & Monitoring Service supports auditing of many different UNIX, Linux, and Windows operating systems.

In Unix and Linux agents, DirectControl Agent is a pre-requisite for the Auditing & Monitoring service.

This release note updates information available in the DirectAudit Administrator's Guide and describes known issues. You can obtain information about previous releases from the Delinea Support Portal, in the Product Documentation page.

Feature Changes in Auditing & Monitoring Service 5.9.1 (Release 2022.1)

General

Compatibility

- With the Server Suite Agent for Windows version 19.6 and later, the Audit and Monitoring Service uses a different compression library to compress the video data being sent from the agent to the collector. As a result, this agent and all future versions of agents are *not* compatible with audit collector versions 18.11 or earlier.

  IMPORTANT: You will lose video data if you deploy the newer agents in an environment with 18.11 or older collectors.___ Audit trail events and indexed events lists are not affected in this situation.

Because of this incompatibility and risk of data loss, you MUST upgrade all of your collectors to the 19.6 or higher version BEFORE you upgrade the agents to Release 2021.1.

As a reminder, before you upgrade the collectors you must first upgrade the database schema. So, to summarize, here's the order in which you upgrade the audit components:

1. Upgrade the database.
2. Upgrade the collectors.
3. Upgrade the agents.

- The minimum DirectControl Agent for *NIX version required by this version of the service is 5.9.0 (Release 2022)

Security Fix

N/A

Audit Collector

N/A

Audit Analyzer and Session Player

N/A

Audit Manager

N/A

DirectAudit Agent for

N/A

Database

N/A

FindSessions Tool

N/A

Server Suite Agent for Windows

N/A

Audit Module for PowerShell

- A new '-Limit' parameter was added to the 'Audit Module for Powershell' command 'Get-CdaAuditEvent'. The '-Limit' parameter is used to specify the number of database entries to return in the results. The '-Limit' parameter is an optional parameter. If the parameter is not specified the command will return 65,536 entries which is the same number of entries this command returned before the change. (Ref:430373)

Audit Management Server

N/A

Supported Platforms

For the list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite release notes.

For the platforms to be removed support in coming releases, refer to the "Notice of Termination Support" section in the Server Suite release notes.

Bugs Fixed in this Release

General

Windows Install / Upgrade / Uninstall

Audit Collector

Audit Analyzer and Session Player

Audit Manager

DirectAudit Agent for

- Fixed an issue for AIX where the user login might take minutes on a system that has many running processes. (Ref:453879)

Database

FindSessions Tool

Server Suite Agent for Windows

Audit Module for PowerShell

Audit Management Server

Known Issues

The following sections describe known issues, suggestions, and limitations associated with the Audit and Monitoring Service.

General

For the most up-to-date list of known issues, refer to the knowledge base articles in the Delinea Support Portal.

- Starting in Release 2016, only ADMX format for group policies will be installed and ADM format will no longer be provided. (Ref: CS-6821)
- Starting in Release 2016, Server Suite will no longer be adding new features to the DirectManage Audit SDK component. It is recommended that all existing users of this component start using the Audit Module for PowerShell component, which is the intended replacement of the SDK. (Ref: CS-6713)
- From Release 2017.1 onward, DirectAudit no longer supports Version 1 Audit Store databases. You will no longer be able to attach Version 1 databases to an existing DirectAudit installation. To view data from version 1.x databases, please install a DirectAudit Auditor Console 1.x and attach the database. (Ref: CS-41219)

Windows Install / Upgrade / Uninstall

- If a DirectManage Audit installation has been configured with multiple Audit Management Servers and some of the servers are running on an older version, the Audit Manager may not list these older servers because the new servers list supersedes the older ones. (Ref: CS-40818)
- When upgrading DirectAudit in Windows, you should use the autorun program to perform the upgrade. The autorun program automatically upgrades other Delinea components such as Delinea Licensing Report. If you upgrade DirectAudit components individually using the Microsoft Installer (msi) and then attempt to use the autorun program to uninstall all components, autorun will only be able to uninstall the Delinea Licensing Report that were upgraded to the latest version. You can remove any remaining components manually using the Add/Remove Programs and Features Control Panel. (Ref: 46293a)
- If you run setup.exe with all DirectAudit components selected for installation on a single computer, the operation is known as the "Easy Install." Although this is the default for new installations, using the "Easy Install" option requires you to have local administrator privileges.
- If you uninstall the collector component on a computer that is not joined to the domain, you will see the following messages during an uninstall operation:

The specified domain either does not exist or could not be contacted.

(Exception from HRESULT: 0x8007054B)

Despite the alert message, the collector is successfully uninstalled when you click OK.

Collector

- In the Collector Configuration wizard, if the account credentials you give for the SQL Server do not match an existing account on the SQL Server, and you have the rights to create SQL Server accounts, the credentials you give will be used to automatically create a new SQL Server account.

Audit Analyzer and Session Player

- Release 2017.3 has introduced a new version of dzdo and PAM authentication audit trail events. However, these events cannot be captured by older version of database/Collector or reported by older versions of DirectAudit Audit Analyzer console or FindSessions utility or PowerShell cmdlets. To rectify this issue, you need to upgrade the DirectAudit backend components (such as Audit Manager console, Audit Analyzer console, Collector, and Audit Store databases) to Release 2017.3 or later version. Contact Delinea support if you are unable to upgrade the DirectAudit backend components so that DirectAudit database patching scripts can be provided to you based on your current version. (Ref: CS-44654)
- When detaching and re-attaching an Audit Store database from an Audit Store, Delinea recommends refreshing the query results for all open queries in Audit Analyzer console prior to replaying a session from that database. Failure to do so may result into a database error. (Ref: CS-42125)
- If the active audit store database spans two SQL databases, the Audit Analyzer will show UNIX sessions as "Disconnected" until some data is received from those sessions. Once data has been received, the session state will change to "In Progress."
- If an audited Windows session is using multiple monitors in extended mode in DirectAudit 3.2.2 or earlier, it cannot be exported as WMV files. In DirectAudit 3.2.3 or later, it will be trimmed to 2048x2048 pixels before it is saved and can be exported as in WMV file in 2048x2048 resolution. (Ref: 27003a, 75163, CS-6450, CS-3265).
- When Server Suite Agent for Windows machine's system color depth is changed during an audited session, the playback of the session may not be displayed properly. (Ref: 36818c)

- Entering specific keywords in the “Application” Event list column will not filter based on the keywords as expected. For example, entering the search term "c" will locate the string "Windows Explorer". This is because application characteristics are stored in the database as a set of related attributes as follows: "Explorer.EXE | Microsoft® Windows® Operating System | Windows Explorer | Microsoft Corporation | 6.1.7600.16385" A match with any of the Windows Explorer attributes will yield “Windows Explorer". This issue will be addressed in an upcoming release. (Ref: 39645b)
- In Audit Analyzer, you can specify double-quote enclosed strings in the query that searches for “Unix Commands and Outputs” attribute. However, if a double-quote character is inside the double-quote enclosed string, the query result is undefined. (Ref: CS-39348)
- If a DirectAudit Installation is configured to not capture video data, parameters of the UNIX command are also not captured. Therefore, the query using "Parameters of Commands and Applications” as the criteria does not work under this configuration. This is a known issue and will be addressed in future release. (Ref: 55741b)
- If you open Audit Analyzer and right click on any child node of predefined queries such as "All, Grouped by User", "All, Grouped by Machine" or "All, Grouped by Audit Store" in the left pane, the context menu is displayed and it shows a menu item named "Properties". This context menu item, when clicked, does not open any dialog box because it is not a valid action for the selected child node. This menu item will be removed in the future release. (Ref: 48681b)
- By default, Audit Analyzer uses MSS2 codec to export audited sessions to a WMV (Windows Media Video) file. The MSS2 codec has a known issue which results in fuzzy video when an audited Windows session is exported as WMV file and opened in Windows Movie Maker 2012. From DirectAudit 3.2.0 onward, you can specify your own codec to export an audited session to a WMV file. Please refer to KB-4029 for additional information. (Ref: 56021a)

Audit Manager

- User and group criteria should not be combined in an Audit Role or it may result into inconsistent results, the workaround is for users to use two different audit roles (one for groups, another for users) if they want to mix users and groups in audit role assignment. (Ref: CS-38968)
- When creating an AuditRole with "ClientName" Audit Manager's Role Properties / Criteria will display an empty value rather than "ClientName = <IP address>" (Ref: CS-41803)
- If you assign DirectAudit permissions to a Domain Local group, which is not in the current domain in the Audit Manager Installation Property Security tab, and a user belonging to that group runs Audit Analyzer and tries to connect to the DirectAudit Installation, Audit Analyzer will display the warning “You do not have permission to connect to the SQL server.” A workaround is to grant permission to a Global or Universal group instead. (Ref: 25546c)

Server Suite DirectAudit Agent for

General

- Delinea recommends customers use the session auditing capability of DirectAudit to ensure the complete login session is audited vs. auditing individual commands. When the administrator configures Direct Audit to audit a specific command, Direct Audit moves the original command executable to a different location and replaces it by a symbolic link to the Direct Audit shell. It is possible for a user to find out the new location of the executable and runs that command directly to bypass auditing. Whereas the likelihood of this happening is very minute, Delinea recommends session auditing be turned on to avoid the chance of this happening.

- If a user is logged in to AIX and HP-UX via a GUI, for example Xmanager, a terminal opened in the GUI will not be audited. To workaround this issue, set the centrifyda.conf parameter 'dash.allinvoked' to true. (Ref: 66330, CS-5876)

- Obfuscation of session data has the following limitation: If the information is sent to stdout not as a whole, but piece by piece, the information will not be obfuscated. Example: A user wants to obfuscate a pattern "1234-5678". However, "1234-" is shown first and "5678" is shown 1 second later, this pattern will not be obfuscated. Since the stdout buffer in the audit shell is 4KB, the obfuscation string is at most 4KB long. Note: this applies to stdout only. (80462a)

- Auditing init during startup on UNIX is not possible. The init command used during the boot process should not be audited using per-command auditing. If you attempt to audit init, your operating system will not reboot properly.

- You cannot start a GUI session if you are logged in via an interactive session. Running startx or starting a GUI session from an interactive session results in the following message:

  X: user not authorized to run the X server, aborting.

  Workaround:

  - Run "sudo dpkg-reconfigure x11-common"

  - When you are prompted for users allowed to start the X server, choose "anybody" (the default is "console users only").

  The GUI session or X server should start normally. (Ref: 25036a)

- To audit the GUI terminal emulators, GUI login managers have to be fully reinitialized after auditing is enabled. On Linux, "init 3 && init 5" will start the reinitialization. (Stopping the X server only, or pressing ctrl+alt+backspace in Gnome, will not start the reinitialization.)
- When a local user and an Active Directory user use the same UNIX user name, the user name will default to the name of the Active Directory user. If the local user name is intended, setting the pam.allow.override parameter in /etc/centrifydc/centrifydc.conf will help. After this setting, the user name implies the Active Directory user; and <username>@localhost will imply the local user.

  DirectAudit 3.0 or later understands the "@localhost" syntax. DirectControl Agent will respond to <username>@localhost if the user name is set in pam.allow.override.

  If you upgrade from DirectAudit 2.0, disable DirectAudit so that the new DirectAudit mechanism for hooking shells can be installed: Run 'dacontrol –d -a' to disable auditing, then restart the upgrade.

  DirectAudit maintains a cache of user information for performance reasons. This cache interferes with Unix commands that manipulate the local user database (passwd file). These commands include useradd, userdel and usermod. From DirectAudit 3.2.0 onwards, DirectAudit will not access its local cache to fully support the following commands: useradd, userdel, adduser, usermod, mkuser, rmuser, chuser

  Please contact support if your operating system platform has other programs that directly access the local passwd file. (Ref: 56259a)
- If session auditing is enabled, all local user logins are processed by DirectAudit to determine whether the session should be audited. This may block login if domain controllers are not responsive and/or DirectControl Agent is not running. Two new parameters are introduced in /etc/centrifyda/centrifyda.conf:

  - user.ignore: specifies a list of local users that DirectAudit does not use Active Directory to determine audit level. By default, the list is /etc/centrifydc/user.ignore (the same one that DirectControl uses), which includes some important accounts like root, bin, daemon, etc.

  - user.ignore.audit.level - specifies the audit level for the local users specified in the user.ignore list. The supported values are 0 (audit if possible) and 1 (audit not requested/required). Default is 0 (audit if possible). Note that "audit required" is not a reasonable choice, as this user needs to login all the time; and "audit required" may block login if DirectAudit does not function correctly. (Ref: 55599a, 57946a, 56935a, 58251a)
- The /usr/share/centrifydc/bin/centrifyda script should be used to start/stop DirectAudit service in all *nix platforms. However, systemd is not fully supported in /usr/share/centrifydc/bin/centrifyda. For platforms that use systemd by default (such as SUSE Linux Enterprise 12/SUSE Linux Desktop 12), users need to set the environment variable SYSTEMD_NO_WRAP to 1 before calling the /usr/share/centrifydc/bin/centrifyda. Operations such as killing a daemon, running dad (DirectAudit daemon) directly, or running dastop command, could lead to issues in daemon managers in some *nix platforms. For example, SMF of Solaris, SRC of AIX and systemd of Fedora 20, may record incorrect running status of the daemon; and may fail to start daemon. (Ref: 57653a, 71211a)
- Disable auditing before upgrade

  If you upgrade from DirectAudit 2.0, please run "dacontrol -d -a" to disable DirectAudit before upgrade. Both the installer shell script, install-da.sh, and the native package manager will detect if auditing is enabled and abort if so.

  If you are using the native package manager to upgrade and youattempt to upgrade while auditing is enabled, you may find that,after the package manager aborts, the DirectAudit installation isshown as broken. This may be ignored. Simply disable auditing,upgrade and then re-enable auditing and the package will beshown as committed.

RedHat Linux

- Due to a limitation of some implementations of audispd (audit dispatcher daemon provided by the operating system), DirectAudit advanced monitoring feature may not work if “dacontrol –n/-m” was run multiple times and over the limit specified in the parameter max_restarts in /etc/audisp/audispd.conf (default 10). If you enable the DirectAudit Advanced monitoring feature and it does not generate the audit trail events as expected, you can run dainfo to check on the status of advanced monitoring feature. If the program /usr/share/centrifydc/bin/dadispatcher is not running, dainfo will show “DirectAudit advanced monitoring status” as “not running”. In this case, you need to restart the system audit daemon using the command “service auditd restart”. This will re-activate the advanced monitoring feature. (Ref: CS-41267)
- The characters ('%', '#', '>' and '$') are used by DirectAudit to recognize UNIX commands. They should not be used in role names and as part of trouble-tickets; otherwise they will be recognized as part of a UNIX command. (Ref: 51687a)
- DirectAudit advanced monitoring features may not work with early versions of RedHat 5 due to different system configurations. The earliest version that Delinea tested is RedHat 5.6. Please contact Delinea Support if you need support in versions earlier than RedHat 5.6. (Ref: CS-43042)
- The advanced monitoring feature in RedHat 5 version only supports selinux mode set to 'disabled' or 'permissive', 'enforcing' is not supported due to incompatible selinux policies. Moreover, advanced monitoring feature may not work with earlier versions of RedHat 5 releases due to different system configurations. Please contact Delinea support if you need support in versions earlier than RedHat 5.6. (Ref: CS-43024)

Debian Linux

- To install the Delinea DirectAudit package on a computer with the Debian operating environment, you must use the dpkg --install or dpkg -i option. You cannot use the dpkg --update or dpkg -u options to install or update the Delinea DirectAudit package. If you need to update the Delinea DirectAudit package, you need to first delete the old package using the dpkg --purge or dpkg -P option then install the new package with the dpkg --install or dpkg -i option.

  > **Note**: Do not use the dpkg --remove or dpkg -r command to remove Delinea DirectAudit. Using the --remove option prevents the DirectAudit configuration file, /etc/centrifyda/centrifyda.conf, from being created properly when you reinstall the package.

Solaris

- Delinea recommends that you install the appropriate recommended patch bundles for the version of Sun Solaris you are using before installing Delinea DirectAudit.

  The patch installation will skip any individual patches that don't apply to your system, and you can use Sun's patch management system to ensure your computers get the latest security fixes.

  To help you identify any required patches for your environment, Delinea supplies the pca patch checker in all Solaris Delinea Server Suite packages. Install.sh will prompt you to check the patch level of your environment during installation.

  To check for Sun recommended patches with the pca patch checker you should have the wget package installed. This package may be obtained from:

  http://ftp.wayne.edu/sun_freeware/

  And source code may be obtained from:

  http://www.gnu.org/software/wget/

  For more information about downloading and installing patches, see the Sun Web site.

  The minimum patches required for Delinea DirectAudit are provided below for reference purposes. In some cases these patches may be obsoleted or incorporated into other patches, so the patch numbers on your Solaris machines may be different. The authoritative source on patch compatibility is Sun; their Web site will allow you to follow patch histories to ensure any later patches you are using are compatible with the ones required by DirectAudit.

  For Solaris 10: 119254-65 120011-14 127127-11 138263-03

- Please contact technical support if you are using sparse zone(s) and like to do one of the following:
  - Change session auditing status from disabled to enabled during upgrade.
  - Enable session auditing in a global zone and want to disable session auditing in sparse zone(s) when using the same global zone. (Ref: 76572, 80616b)

- The following commands, located in /usr/bin, might be implemented as ksh programs or scripts:

  alias bg cd

  command fc fg

  getopts hash jobs

  kill read test

  type ulimit umask

  unalias wait

  To identify commands implemented as ksh scripts, run the following script:

  #!/bin/ksh -p

  cmd=`basename $0`

```
$cmd "$@"
```

The commands that are implemented internally by ksh should not be audited.

- On a system using SMF (Service Management Facility), such as Solaris 10, the DirectAudit daemon might not start up after an upgrade from DirectAudit 1.x. This does not affect a fresh installation. To bring the daemon up, run these commands:
  - svcadm disable centrifyda
  - svcadm enable centrifyda
  - Run 'svcs' and find 'centrifyda' to confirm the daemon is online.

AIX

- Some versions of AIX sshd do not function reliably with Delinea products. When possible, Delinea recommends using sshd included in Delinea openSSH on AIX platforms. (Ref: CS-7098)
- Local AIX users cannot be audited when they log in via built-in ssh, due to a change in AIX 7.0 ML1. Customers are advised to install Delinea OpenSSH if auditing of ssh login by local users is required (Ref: 33299a).
- Change in AIX root user behavior: By default, all releases starting with Release 2014 (DirectAudit 3.2.0) DO NOT modify the root stanza in AIX for new installations. One side effect is that root user login WILL NOT be audited. If your environment requires session auditing of root user login, you need to do the followings:
  a. Set up a DirectAuthorize role that has the audit level of "audit required" or "audit if possible"; and assign this role to root.
  b. Set the parameter adclient.autoedit.user.root to TRUE in /etc/centrifydc/centrifydc.conf.
  c. If DirectAudit session auditing is not enabled, enable DirectAudit session auditing using the command "dacontrol -e".
  d. Restart adclient (Ref: 56239a, 56604a)
- For AIX customers who upgrade from prior versions of Release 2014 (DirectAudit 3.2.0), there is NO change in behavior. The parameter adclient.autoedit.user.root is set to true in /etc/centrifydc/centrifydc.conf. The root user will still be audited. (Ref: 56235)

HPUX

You can install this package by copying it to a HP-UX computer and running install.sh, the installer, or by running the following commands, where <release> is the version of the DirectAudit package you are installing:

```
gzip -d centrifyda-<release>-hp11.31-ia64.depot.gz

swinstall -s /path/centrifyda-<release>-hp11.31-ia64.depot \

-x allow_incompatible=true
```

- You must specify the full path to the Delinea DirectAudit depot file and set the allow_incompatible option to true to install successfully.
- The installation script checks your environment for the minimum patch levels required. If you have more recent patches installed, however, you may see an error message. To install, re-run the installation command with the following additional command line option:

  ```
  -x enforce_scripts=false
  ```

Database

- When adding an Audit Store database to a SQL Server Availability Group with the multi subnet failover feature, the SQL Server that hosts the management database must be SQL Server 2012 or above. In addition, when upgrading an existing DirectAudit installation to use the SQL Server Availability Group feature, Delinea recommends upgrading Collectors, Audit Management Server service, Audit Manager consoles and Audit Analyzer consoles to the latest version to benefit from this feature. (Ref: CS-39872)
- In previous versions of DirectAudit, it was possible to specify the location of the database file. In DirectAudit 2.0.0 and later this capability is not provided in the Audit Store Database Wizard. However, you can still specify the full text file location, database file location, or transaction log file location by choosing "View SQL Scripts" and modifying the relevant database location manually in the script.
- If the default memory setting for SQL Server is more than the actual memory in the system a memory error may occur. For more information see:

http://social.msdn.microsoft.com/Forums/en-US/sqldatabaseengine/thread/74a94f06-adf5-4059-bb92-57a99def37bd/

SQL Server 2008 R2 full text search categorizes certain words as stop words by default and ignores them for searches. Some stop words are common UNIX commands such as like, which, do, and while. For more details about stop words and how to configure, please refer to http://technet.microsoft.com/en-us/library/ms142551.aspx

- The collector monitors the active Audit Store database to check if it is running low on disk space. If an active Audit Store the database is on a disk with volume mount point, the collector may give a false alarm. In such cases, it is recommended to disable the detection by setting the following registry key with the type of DWORD to 0 on all your collector machines. (Ref: 53389a)

  HKLM\Software\Centrify\DirectAudit\Collector\AuditStoreDiskSpaceLowThreshold

- Collector only detects AuditStore disk space low against a configurable threshold if the SQL Server version is 2008 R2 SP1 (10.50.2500.0) and above. The threshold can be configured at Collector machine Registry: HKLM\Software\Centrify\DirectAudit\Collector\AuditStoreDiskSpaceLowThreshold DWORD in MB, not configured, default to 1024 MB. If free disk space is less than the threshold, Collector state is changed to "AuditStore database disk space is low", and stops accepting audit data from Agent(s).

Audit Management Server

- To configure the audit management server to point to an installation, the user who is running the Audit Management Server Configuration Wizard must have the "Manage SQL Logins" permission on the management database of the installation. For example, if you are configuring an audit management server in an external forest with a one-way trust, be sure that the installation supports Windows and SQL Server authentication and the account you are using is from the internal forest and has the "Manage SQL Logins" permission on the management database. (Ref: 46989a)

FindSession Tools

- For per-command auditing of dzdo command, when a ticket is entered, the role and ticket are associated with the audited session. For such sessions, the FindSessions tool's export of type UnixCommand, UnixInput, or UnixInputOutput based on the role and/or ticket criteria will have the exported command, STDIN, or STDIN and STDOUT marked with role and ticket. When per session auditing is enabled, the exported data will not have role and ticket information. (Ref: 53936a)

- When per-command auditing is enabled for dzdo command, and role and trouble ticket capturing is also configured, FindSessions.exe run with /export=UnixCommand option will not show the role and trouble ticket information in the exported file for the dzdo command itself, if the dzdo command executed is "dzdo su –" or "dzdo –i". However, all the command executed within that dzdo session will have correct role and trouble ticket information. (Ref: 51787a)

Server Suite Agent for Windows

- When a user disconnects and then later reconnects to an existing user session from a switch user operation, a successful logon audit trail message will not be logged after the user has reconnected to the session though authentication. This does not apply when the user is performing lock and unlock operations or the logon method is different from the previous login (remote vs. console logon). (Ref: CS-41453)

- In the DirectAudit Agent for Windows control panel, the setting "Maximum size of the offline data file" indicates the minimum amount of disk space (in percentage) that must be available/free in the spool volume in order to continue auditing users (especially when the DirectAudit Agent cannot send audit data to collector). The DirectAudit Agent makes its best attempt to pause auditing when the specified amount of disk space is no longer available and in certain cases may continue to write to spool volume for a few minutes before eventually pausing the auditing activity. (78072, CS-6718)

- The optional video capture feature requires both the Collector and the DirectAudit Agent to use 2013.2 or later. If any of collectors or agents are running an older version, video data may still be recorded even though you have turned it off in Release 2013 Update 2 Audit Manager. (Ref: 44064a)

- If Server Suite Agent for Windows is auditing a Windows 8 or Windows 2012 system, the Indexed Event List of the corresponding audited session will not show any events for the applications that are using the Metro User Interface. The Metro UI is not supported. (Ref: 56556b)

- Upon making changes to Group Policy "Centrify Audit Trail Setting" > "Centrify Common Setting" > "Send audit trail to log file", it would require reboot of the client computer (agent) for this setting to be effective despite the Group Policy has already been refreshed on the client computer. (Ref: 73368b)

- The offline data location (and subdirectories below it) is expected to be a location dedicated to spooling, for example c:\spool. If the offline data location is changed, all files in the old location (including subdirectories and their contents) are moved to the new location. This may cause problems if the old location was not exclusively for spooling use. For example, choosing c:\ as the original spool location and d:\spool as the new location would cause all files on the c:\ drive to be copied to d:\spool. (Ref: 26592a)

- Some events related to the login script are not listed in the indexed events list. The login script cannot be audited for an initial few seconds because the

DirectAudit software has not completed its setup. (Ref: 26286a)

- Some events related to the login script are not listed in the indexed events list. The login script cannot be audited for an initial few seconds because the Server Suite Agent for Windows software has not completed its setup. (Ref: 26286a)

Delinea Audit Module for PowerShell

- Audit Module for PowerShell may take a long time to start because of the publisher's certificate verification. To resolve the problem, disable the "Check for publisher's certificate revocation" option in System Control Panel\Internet Options\Advanced\Security. (Ref: 72499)
- After installing Audit Module for PowerShell in a RDP session, PowerShell complains module "Delinea.DirectAudit.PowerShell" cannot be loaded. This is because the installation package needs to modify system environment variables to let PowerShell know where to load the module. This operation needed to be done in a "Console Session" if installation is done via RDP. To resolve this problem, logout and re-login or run RDP with the "admin" option as "mstsc /admin" or "mstsc /console". (Ref: 72500a)

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

About Server Suite Auditing & Monitoring Service

Delinea Server Suite is a product category that includes the following product offerings:

- Privileged Access Service
- Authentication Service
- Privilege Elevation Service
- Auditing & Monitoring Service

The DirectControl Agent provides services for the Authentication Service and Privilege Elevation Service contained in the CentrifyDC packages. The DirectAudit Agent provides services for Auditing & Monitoring Service contained in the CentrifyDA packages.

The Auditing & Monitoring Service is a key component of Server Suite. It enables detailed auditing of user activity on a wide range of UNIX, Linux, and Windows computers. With this service, you can perform immediate, in-depth troubleshooting by replaying user activity that may have contributed to system failures, spot suspicious activity by monitoring current user sessions, improve regulatory compliance, and ensure accountability by capturing and storing detailed information about the applications used and the commands executed. If you enable auditing, the Server Suite Agent for Windows records user activity on the Windows computer when it is installed. Auditing & Monitoring Service supports auditing of many different UNIX, Linux, and Windows operating systems.

In Unix and Linux agents, DirectControl Agent is a pre-requisite for the Auditing & Monitoring service.

Starting in Release 2016, only ADMX format for group policies will be installed and ADM format will no longer be provided. (Ref: CS-6821)

Starting in Release 2016, Server Suite will no longer be adding new features to the DirectManage Audit SDK component. It is recommended that all existing users of this component start using the Audit Module for PowerShell component, which is the intended replacement of the SDK. (Ref: CS-6713)

From Release 2017.1 onward, DirectAudit no longer supports Version 1 Audit Store databases. You will no longer be able to attach Version 1 databases to an existing DirectAudit installation. To view data from version 1.x databases, please install a DirectAudit Auditor Console 1.x and attach the database. (Ref: CS-41219)

This release note updates information available in the DirectAudit Administrator's Guide and describes known issues. You can obtain information about previous releases from the Delinea Support Portal, in the Product Documentation page.

Delinea software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962. (Ref: CS-44575)

Feature Changes in Auditing & Monitoring Service 5.9.0 (Release 2022)

General

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Compatibility

- With the Server Suite Agent for Windows version 19.6 and later, the Audit and Monitoring Service uses a different compression library to compress the video data being sent from the agent to the collector. As a result, this agent and all future versions of agents are *not* compatible with audit collector versions 18.11 or earlier.

  IMPORTANT: You will lose video data if you deploy the newer agents in an environment with 18.11 or older collectors.___ Audit trail events and indexed events lists are not affected in this situation.

Because of this incompatibility and risk of data loss, you MUST upgrade all of your collectors to the 19.6 or higher version BEFORE you upgrade the agents to Release 2021.1.

As a reminder, before you upgrade the collectors you must first upgrade the database schema. So, to summarize, here's the order in which you upgrade the audit components:

1. Upgrade the database.
2. Upgrade the collectors.
3. Upgrade the agents.

- The minimum DirectControl Agent for *NIX version required by this version of the service is 5.9.0 (Release 2022)

Security Fix

N/A

Audit Collector

N/A

Audit Analyzer and Session Player

N/A

Audit Manager

N/A

DirectAudit Agent for

N/A

Database

N/A

FindSessions Tool

N/A

Server Suite Agent for Windows

N/A

Audit Module for PowerShell

N/A

Audit Management Server

N/A

Supported Platforms

For the list of the supported platforms by this release, refer to the "Supported Platforms" section in the Server Suite release notes.

For the platforms to be removed support in coming releases, refer to the "Notice of Termination Support" section in the Server Suite release notes.

Bugs Fixed in this Release

General

Windows Install / Upgrade / Uninstall

Audit Collector

Fixed an issue where commands longer than 255 characters were truncated. You can now set a new Registry value HKLM\SOFTWARE\Centrify\DirectAudit\Collector\StdinBuffSize so that large commands can be logged. The default value is 255.

Audit Analyzer and Session Player

Audit Manager

DirectAudit Agent for

Fixed an issue where the third-party program might crash in DirectAudit NSS/LAM module when session auditing is enabled.

Database

FindSessions Tool

Server Suite Agent for Windows

Audit Module for PowerShell

Audit Management Server

Known Issues

The following sections describe known issues, suggestions, and limitations associated with the Audit and Monitoring Service.

General

For the most up-to-date list of known issues, refer to the knowledge base articles in the Delinea Support Portal.

Windows Install / Upgrade / Uninstall

- If a DirectManage Audit installation has been configured with multiple Audit Management Servers and some of the servers are running on an older version, the Audit Manager may not list these older servers because the new servers list supersedes the older ones. (Ref: CS-40818)
- When upgrading DirectAudit in Windows, you should use the autorun program to perform the upgrade. The autorun program automatically upgrades other Delinea components such as Delinea Licensing Report. If you upgrade DirectAudit components individually using the Microsoft Installer (msi) and then attempt to use the autorun program to uninstall all components, autorun will only be able to uninstall the Delinea Licensing Report that were upgraded to the latest version. You can remove any remaining components manually using the Add/Remove Programs and Features Control Panel. (Ref: 46293a)
- If you run setup.exe with all DirectAudit components selected for installation on a single computer, the operation is known as the "Easy Install." Although this is the default for new installations, using the "Easy Install" option requires you to have local administrator privileges.
- If you uninstall the collector component on a computer that is not joined to the domain, you will see the following messages during an uninstall operation:

The specified domain either does not exist or could not be contacted.

(Exception from HRESULT: 0x8007054B)

Despite the alert message, the collector is successfully uninstalled when you click OK.

Collector

- In the Collector Configuration wizard, if the account credentials you give for the SQL Server do not match an existing account on the SQL Server, and you have the rights to create SQL Server accounts, the credentials you give will be used to automatically create a new SQL Server account.

Audit Analyzer and Session Player

- Release 2017.3 has introduced a new version of dzdo and PAM authentication audit trail events. However, these events cannot be captured by older version of database/Collector or reported by older versions of DirectAudit Audit Analyzer console or FindSessions utility or PowerShell cmdlets. To rectify this issue, you need to upgrade the DirectAudit backend components (such as Audit Manager console, Audit Analyzer console, Collector, and Audit Store databases) to Release 2017.3 or later version. Contact Delinea support if you are unable to upgrade the DirectAudit backend components so that DirectAudit database patching scripts can be provided to you based on your current version. (Ref: CS-44654)
- When detaching and re-attaching an Audit Store database from an Audit Store, Delinea recommends refreshing the query results for all open queries in Audit Analyzer console prior to replaying a session from that database. Failure to do so may result into a database error. (Ref: CS-42125)
- If the active audit store database spans two SQL databases, the Audit Analyzer will show UNIX sessions as "Disconnected" until some data is received from those sessions. Once data has been received, the session state will change to "In Progress."

- If an audited Windows session is using multiple monitors in extended mode in DirectAudit 3.2.2 or earlier, it cannot be exported as WMV files. In DirectAudit 3.2.3 or later, it will be trimmed to 2048x2048 pixels before it is saved and can be exported as in WMV file in 2048x2048 resolution. (Ref: 27003a, 75163, CS-6450, CS-3265).
- When Server Suite Agent for Windows machine's system color depth is changed during an audited session, the playback of the session may not be displayed properly. (Ref: 36818c)
- Entering specific keywords in the "Application" Event list column will not filter based on the keywords as expected. For example, entering the search term "c" will locate the string "Windows Explorer". This is because application characteristics are stored in the database as a set of related attributes as follows: "Explorer.EXE | Microsoft® Windows® Operating System | Windows Explorer | Microsoft Corporation | 6.1.7600.16385" A match with any of the Windows Explorer attributes will yield "Windows Explorer". This issue will be addressed in an upcoming release. (Ref: 39645b)
- In Audit Analyzer, you can specify double-quote enclosed strings in the query that searches for "Unix Commands and Outputs" attribute. However, if a double-quote character is inside the double-quote enclosed string, the query result is undefined. (Ref: CS-39348)
- If a DirectAudit Installation is configured to not capture video data, parameters of the UNIX command are also not captured. Therefore, the query using "Parameters of Commands and Applications" as the criteria does not work under this configuration. This is a known issue and will be addressed in future release. (Ref: 55741b)
- If you open Audit Analyzer and right click on any child node of predefined queries such as "All, Grouped by User", "All, Grouped by Machine" or "All, Grouped by Audit Store" in the left pane, the context menu is displayed and it shows a menu item named "Properties". This context menu item, when clicked, does not open any dialog box because it is not a valid action for the selected child node. This menu item will be removed in the future release. (Ref: 48681b)
- By default, Audit Analyzer uses MSS2 codec to export audited sessions to a WMV (Windows Media Video) file. The MSS2 codec has a known issue which results in fuzzy video when an audited Windows session is exported as WMV file and opened in Windows Movie Maker 2012. From DirectAudit 3.2.0 onward, you can specify your own codec to export an audited session to a WMV file. Please refer to KB-4029 for additional information. (Ref: 56021a)

Audit Manager

- User and group criteria should not be combined in an Audit Role or it may result into inconsistent results, the workaround is for users to use two different audit roles (one for groups, another for users) if they want to mix users and groups in audit role assignment. (Ref: CS-38968)
- When creating an AuditRole with "ClientName" Audit Manager's Role Properties / Criteria will display an empty value rather than "ClientName = <IP address>" (Ref: CS-41803)
- If you assign DirectAudit permissions to a Domain Local group, which is not in the current domain in the Audit Manager Installation Property Security tab, and a user belonging to that group runs Audit Analyzer and tries to connect to the DirectAudit Installation, Audit Analyzer will display the warning "You do not have permission to connect to the SQL server." A workaround is to grant permission to a Global or Universal group instead. (Ref: 25546c)

Server Suite DirectAudit Agent for

General

- Delinea recommends customers use the session auditing capability of DirectAudit to ensure the complete login session is audited vs. auditing individual commands. When the administrator configures Direct Audit to audit a specific command, Direct Audit moves the original command executable to a different location and replaces it by a symbolic link to the Direct Audit shell. It is possible for a user to find out the new location of the executable and runs that command directly to bypass auditing. Whereas the likelihood of this happening is very minute, Delinea recommends session auditing be turned on to avoid the chance of this happening.

- If a user is logged in to AIX and HP-UX via a GUI, for example Xmanager, a terminal opened in the GUI will not be audited. To workaround this issue, set the centrifyda.conf parameter 'dash.allinvoked' to true. (Ref: 66330, CS-5876)

- Obfuscation of session data has the following limitation: If the information is sent to stdout not as a whole, but piece by piece, the information will not be obfuscated. Example: A user wants to obfuscate a pattern "1234-5678". However, "1234-" is shown first and "5678" is shown 1 second later, this pattern will not be obfuscated. Since the stdout buffer in the audit shell is 4KB, the obfuscation string is at most 4KB long. Note: this applies to stdout only. (80462a)

- Auditing init during startup on UNIX is not possible. The init command used during the boot process should not be audited using per-command auditing. If you attempt to audit init, your operating system will not reboot properly.

- You cannot start a GUI session if you are logged in via an interactive session. Running startx or starting a GUI session from an interactive session results in the following message:

  X: user not authorized to run the X server, aborting.

  Workaround:

- Run "sudo dpkg-reconfigure x11-common"

- When you are prompted for users allowed to start the X server, choose "anybody" (the default is "console users only").

The GUI session or X server should start normally. (Ref: 25036a)

- To audit the GUI terminal emulators, GUI login managers have to be fully reinitialized after auditing is enabled. On Linux, "init 3 && init 5" will start the reinitialization. (Stopping the X server only, or pressing ctrl+alt+backspace in Gnome, will not start the reinitialization.)
- When a local user and an Active Directory user use the same UNIX user name, the user name will default to the name of the Active Directory user. If the local user name is intended, setting the pam.allow.override parameter in /etc/centrifydc/centrifydc.conf will help. After this setting, the user name implies the Active Directory user; and <username>@localhost will imply the local user.

  DirectAudit 3.0 or later understands the "@localhost" syntax. DirectControl Agent will respond to <username>@localhost if the user name is set in pam.allow.override.

  If you upgrade from DirectAudit 2.0, disable DirectAudit so that the new DirectAudit mechanism for hooking shells can be installed: Run 'dacontrol –d -a' to disable auditing, then restart the upgrade.

  DirectAudit maintains a cache of user information for performance reasons. This cache interferes with Unix commands that manipulate the local user database (passwd file). These commands include useradd, userdel and usermod. From DirectAudit 3.2.0 onwards, DirectAudit will not access its local cache to fully support the following commands: useradd, userdel, adduser, usermod, mkuser, rmuser, chuser

  Please contact support if your operating system platform has other programs that directly access the local passwd file. (Ref: 56259a)
- If session auditing is enabled, all local user logins are processed by DirectAudit to determine whether the session should be audited. This may block login if domain controllers are not responsive and/or DirectControl Agent is not running. Two new parameters are introduced in /etc/centrifyda/centrifyda.conf:

  - user.ignore: specifies a list of local users that DirectAudit does not use Active Directory to determine audit level. By default, the list is /etc/centrifydc/user.ignore (the same one that DirectControl uses), which includes some important accounts like root, bin, daemon, etc.

  - user.ignore.audit.level - specifies the audit level for the local users specified in the user.ignore list. The supported values are 0 (audit if possible) and 1 (audit not requested/required). Default is 0 (audit if possible). Note that "audit required" is not a reasonable choice, as this user needs to login all the time; and "audit required" may block login if DirectAudit does not function correctly. (Ref: 55599a, 57946a, 56935a, 58251a)
- The /usr/share/centrifydc/bin/centrifyda script should be used to start/stop DirectAudit service in all *nix platforms. However, systemd is not fully supported in /usr/share/centrifydc/bin/centrifyda. For platforms that use systemd by default (such as SUSE Linux Enterprise 12/SUSE Linux Desktop 12), users need to set the environment variable SYSTEMD_NO_WRAP to 1 before calling the /usr/share/centrifydc/bin/centrifyda. Operations such as killing a daemon, running dad (DirectAudit daemon) directly, or running dastop command, could lead to issues in daemon managers in some *nix platforms. For example, SMF of Solaris, SRC of AIX and systemd of Fedora 20, may record incorrect running status of the daemon; and may fail to start daemon. (Ref: 57653a, 71211a)
- Disable auditing before upgrade

  If you upgrade from DirectAudit 2.0, please run "dacontrol -d -a" to disable DirectAudit before upgrade. Both the installer shell script, install-da.sh, and the native package manager will detect if auditing is enabled and abort if so.

  If you are using the native package manager to upgrade and youattempt to upgrade while auditing is enabled, you may find that,after the package manager aborts, the DirectAudit installation isshown as broken. This may be ignored. Simply disable auditing,upgrade and then re-enable auditing and the package will beshown as committed.

RedHat Linux

- Due to a limitation of some implementations of audispd (audit dispatcher daemon provided by the operating system), DirectAudit advanced monitoring feature may not work if “dacontrol –n/-m” was run multiple times and over the limit specified in the parameter max_restarts in /etc/audisp/audispd.conf (default 10). If you enable the DirectAudit Advanced monitoring feature and it does not generate the audit trail events as expected, you can run dainfo to check on the status of advanced monitoring feature. If the program /usr/share/centrifydc/bin/dadispatcher is not running, dainfo will show “DirectAudit advanced monitoring status” as “not running”. In this case, you need to restart the system audit daemon using the command “service auditd restart”. This will re-activate the advanced monitoring feature. (Ref: CS-41267)
- The characters ('%', '#', '>' and '$') are used by DirectAudit to recognize UNIX commands. They should not be used in role names and as part of trouble-tickets; otherwise they will be recognized as part of a UNIX command. (Ref: 51687a)

- DirectAudit advanced monitoring features may not work with early versions of RedHat 5 due to different system configurations. The earliest version that Delinea tested is RedHat 5.6. Please contact Delinea Support if you need support in versions earlier than RedHat 5.6. (Ref: CS-43042)
- The advanced monitoring feature in RedHat 5 version only supports selinux mode set to 'disabled' or 'permissive', 'enforcing' is not supported due to incompatible selinux policies. Moreover, advanced monitoring feature may not work with earlier versions of RedHat 5 releases due to different system configurations. Please contact Delinea support if you need support in versions earlier than RedHat 5.6. (Ref: CS-43024)

Debian Linux

- To install the Delinea DirectAudit package on a computer with the Debian operating environment, you must use the dpkg --install or dpkg -i option. You cannot use the dpkg --update or dpkg -u options to install or update the Delinea DirectAudit package. If you need to update the Delinea DirectAudit package, you need to first delete the old package using the dpkg --purge or dpkg -P option then install the new package with the dpkg --install or dpkg -i option.

  > **Note**: Do not use the dpkg --remove or dpkg -r command to remove Delinea DirectAudit. Using the --remove option prevents the DirectAudit configuration file, /etc/centrifyda/centrifyda.conf, from being created properly when you reinstall the package.

Solaris

- Delinea recommends that you install the appropriate recommended patch bundles for the version of Sun Solaris you are using before installing Delinea DirectAudit.

  The patch installation will skip any individual patches that don't apply to your system, and you can use Sun's patch management system to ensure your computers get the latest security fixes.

  To help you identify any required patches for your environment, Delinea supplies the pca patch checker in all Solaris Delinea Server Suite packages. Install.sh will prompt you to check the patch level of your environment during installation.

  To check for Sun recommended patches with the pca patch checker you should have the wget package installed. This package may be obtained from:

  http://ftp.wayne.edu/sun_freeware/

  And source code may be obtained from:

  http://www.gnu.org/software/wget/

  For more information about downloading and installing patches, see the Sun Web site.

  The minimum patches required for Delinea DirectAudit are provided below for reference purposes. In some cases these patches may be obsoleted or incorporated into other patches, so the patch numbers on your Solaris machines may be different. The authoritative source on patch compatibility is Sun; their Web site will allow you to follow patch histories to ensure any later patches you are using are compatible with the ones required by DirectAudit.

  For Solaris 10: 119254-65 120011-14 127127-11 138263-03

- Please contact technical support if you are using sparse zone(s) and like to do one of the following:
  - Change session auditing status from disabled to enabled during upgrade.
  - Enable session auditing in a global zone and want to disable session auditing in sparse zone(s) when using the same global zone. (Ref: 76572, 80616b)
- The following commands, located in /usr/bin, might be implemented as ksh programs or scripts:

  alias bg cd

  command fc fg

  getopts hash jobs

  kill read test

  type ulimit umask

  unalias wait

To identify commands implemented as ksh scripts, run the following script:

```
#!/bin/ksh -p

cmd=`basename $0`

$cmd "$@"
```

The commands that are implemented internally by ksh should not be audited.

- On a system using SMF (Service Management Facility), such as Solaris 10, the DirectAudit daemon might not start up after an upgrade from DirectAudit 1.x. This does not affect a fresh installation. To bring the daemon up, run these commands:
  - svcadm disable centrifyda
  - svcadm enable centrifyda
  - Run 'svcs' and find 'centrifyda' to confirm the daemon is online.

AIX

- Some versions of AIX sshd do not function reliably with Delinea products. When possible, Delinea recommends using sshd included in Delinea openSSH on AIX platforms. (Ref: CS-7098)
- Local AIX users cannot be audited when they log in via built-in ssh, due to a change in AIX 7.0 ML1. Customers are advised to install Delinea OpenSSH if auditing of ssh login by local users is required (Ref: 33299a).
- Change in AIX root user behavior: By default, all releases starting with Release 2014 (DirectAudit 3.2.0) DO NOT modify the root stanza in AIX for new installations. One side effect is that root user login WILL NOT be audited. If your environment requires session auditing of root user login, you need to do the followings:
  a. Set up a DirectAuthorize role that has the audit level of "audit required" or "audit if possible"; and assign this role to root.
  b. Set the parameter adclient.autoedit.user.root to TRUE in /etc/centrifydc/centrifydc.conf.
  c. If DirectAudit session auditing is not enabled, enable DirectAudit session auditing using the command "dacontrol -e".
  d. Restart adclient (Ref: 56239a, 56604a)
- For AIX customers who upgrade from prior versions of Release 2014 (DirectAudit 3.2.0), there is NO change in behavior. The parameter adclient.autoedit.user.root is set to true in /etc/centrifydc/centrifydc.conf. The root user will still be audited. (Ref: 56235)

HPUX

You can install this package by copying it to a HP-UX computer and running install.sh, the installer, or by running the following commands, where <release> is the version of the DirectAudit package you are installing:

```
gzip -d centrifyda-<release>-hp11.31-ia64.depot.gz

swinstall -s /path/centrifyda-<release>-hp11.31-ia64.depot \

-x allow_incompatible=true
```

- You must specify the full path to the Delinea DirectAudit depot file and set the allow_incompatible option to true to install successfully.
- The installation script checks your environment for the minimum patch levels required. If you have more recent patches installed, however, you may see an error message. To install, re-run the installation command with the following additional command line option:

  ```
  -x enforce_scripts=false
  ```

Database

- When adding an Audit Store database to a SQL Server Availability Group with the multi subnet failover feature, the SQL Server that hosts the management database must be SQL Server 2012 or above. In addition, when upgrading an existing DirectAudit installation to use the SQL Server Availability Group feature, Delinea recommends upgrading Collectors, Audit Management Server service, Audit Manager consoles and Audit Analyzer consoles to the latest version to benefit from this feature. (Ref: CS-39872)

- In previous versions of DirectAudit, it was possible to specify the location of the database file. In DirectAudit 2.0.0 and later this capability is not provided in the Audit Store Database Wizard. However, you can still specify the full text file location, database file location, or transaction log file location by choosing "View SQL Scripts" and modifying the relevant database location manually in the script.
- If the default memory setting for SQL Server is more than the actual memory in the system a memory error may occur. For more information see:

  http://social.msdn.microsoft.com/Forums/en-US/sqldatabaseengine/thread/74a94f06-adf5-4059-bb92-57a99def37bd/

  SQL Server 2008 R2 full text search categorizes certain words as stop words by default and ignores them for searches. Some stop words are common UNIX commands such as like, which, do, and while. For more details about stop words and how to configure, please refer to http://technet.microsoft.com/en-us/library/ms142551.aspx
- The collector monitors the active Audit Store database to check if it is running low on disk space. If an active Audit Store the database is on a disk with volume mount point, the collector may give a false alarm. In such cases, it is recommended to disable the detection by setting the following registry key with the type of DWORD to 0 on all your collector machines. (Ref: 53389a)

  HKLM\Software\Centrify\DirectAudit\Collector\AuditStoreDiskSpaceLowThreshold
- Collector only detects AuditStore disk space low against a configurable threshold if the SQL Server version is 2008 R2 SP1 (10.50.2500.0) and above. The threshold can be configured at Collector machine Registry: HKLM\Software\Centrify\DirectAudit\Collector\AuditStoreDiskSpaceLowThreshold DWORD in MB, not configured, default to 1024 MB. If free disk space is less than the threshold, Collector state is changed to "AuditStore database disk space is low", and stops accepting audit data from Agent(s).

Audit Management Server

- To configure the audit management server to point to an installation, the user who is running the Audit Management Server Configuration Wizard must have the "Manage SQL Logins" permission on the management database of the installation. For example, if you are configuring an audit management server in an external forest with a one-way trust, be sure that the installation supports Windows and SQL Server authentication and the account you are using is from the internal forest and has the "Manage SQL Logins" permission on the management database. (Ref: 46989a)

FindSession Tools

- For per-command auditing of dzdo command, when a ticket is entered, the role and ticket are associated with the audited session. For such sessions, the FindSessions tool's export of type UnixCommand, UnixInput, or UnixInputOutput based on the role and/or ticket criteria will have the exported command, STDIN, or STDIN and STDOUT marked with role and ticket. When per session auditing is enabled, the exported data will not have role and ticket information. (Ref: 53936a)
- When per-command auditing is enabled for dzdo command, and role and trouble ticket capturing is also configured, FindSessions.exe run with /export=UnixCommand option will not show the role and trouble ticket information in the exported file for the dzdo command itself, if the dzdo command executed is "dzdo su –" or "dzdo –i". However, all the command executed within that dzdo session will have correct role and trouble ticket information. (Ref: 51787a)

Server Suite Agent for Windows

- When a user disconnects and then later reconnects to an existing user session from a switch user operation, a successful logon audit trail message will not be logged after the user has reconnected to the session though authentication. This does not apply when the user is performing lock and unlock operations or the logon method is different from the previous login (remote vs. console logon). (Ref: CS-41453)
- In the DirectAudit Agent for Windows control panel, the setting "Maximum size of the offline data file" indicates the minimum amount of disk space (in percentage) that must be available/free in the spool volume in order to continue auditing users (especially when the DirectAudit Agent cannot send audit data to collector). The DirectAudit Agent makes its best attempt to pause auditing when the specified amount of disk space is no longer available and in certain cases may continue to write to spool volume for a few minutes before eventually pausing the auditing activity. (78072, CS-6718)
- The optional video capture feature requires both the Collector and the DirectAudit Agent to use 2013.2 or later. If any of collectors or agents are running an older version, video data may still be recorded even though you have turned it off in Release 2013 Update 2 Audit Manager. (Ref: 44064a)
- If Server Suite Agent for Windows is auditing a Windows 8 or Windows 2012 system, the Indexed Event List of the corresponding audited session will not show any events for the applications that are using the Metro User Interface. The Metro UI is not supported. (Ref: 56556b)
- Upon making changes to Group Policy "Centrify Audit Trail Setting" > "Centrify Common Setting" > "Send audit trail to log file", it would require reboot of the client computer (agent) for this setting to be effective despite the Group Policy has already been refreshed on the client computer. (Ref: 73368b)

- The offline data location (and subdirectories below it) is expected to be a location dedicated to spooling, for example c:\spool. If the offline data location is changed, all files in the old location (including subdirectories and their contents) are moved to the new location. This may cause problems if the old location was not exclusively for spooling use. For example, choosing c:\ as the original spool location and d:\spool as the new location would cause all files on the c:\ drive to be copied to d:\spool. (Ref: 26592a)
- Some events related to the login script are not listed in the indexed events list. The login script cannot be audited for an initial few seconds because the DirectAudit software has not completed its setup. (Ref: 26286a)
- Some events related to the login script are not listed in the indexed events list. The login script cannot be audited for an initial few seconds because the Server Suite Agent for Windows software has not completed its setup. (Ref: 26286a)

Delinea Audit Module for PowerShell

- Audit Module for PowerShell may take a long time to start because of the publisher's certificate verification. To resolve the problem, disable the "Check for publisher's certificate revocation" option in System Control Panel\Internet Options\Advanced\Security. (Ref: 72499)
- After installing Audit Module for PowerShell in a RDP session, PowerShell complains module "Delinea.DirectAudit.PowerShell" cannot be loaded. This is because the installation package needs to modify system environment variables to let PowerShell know where to load the module. This operation needed to be done in a "Console Session" if installation is done via RDP. To resolve this problem, logout and re-login or run RDP with the "admin" option as "mstsc /admin" or "mstsc /console". (Ref: 72500a)

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

Server Suite 2022 Release Notes

The Delinea Server Suite (previously called Centrify Infrastructure Services, or Centrify Zero Trust Privilege Services) is an integrated family of directory-based authentication, privileged access, privileged elevation, audit & monitoring solutions that secure your cross-platform environment and strengthen regulatory compliance initiatives.

Server Suite includes the following components:

- Authentication Service secures your platforms using the same authentication and Group Policy services deployed for your Windows environment.
- Privilege Elevation Service centrally manages and enforces role-based entitlements for fine-grained control of user access and privileges on UNIX, Linux, and Windows systems.
- Audit & Monitoring Service delivers auditing, logging, and real-time monitoring of user activity on your Windows, UNIX, and Linux system.

This integrated solution helps you improve IT efficiency, strengthen regulatory compliance initiatives, and centrally secure your heterogeneous computing environment.

This release notes cover information specifically about Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service.

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

This release usually includes packages for Windows, UNIX, and Linux operating system environments.

The files for this release are organized onto two media, each available in ISO and zip form:

Server Suite for 64-bit Windows

- main folder

This is the main folder containing information pertinent to this release.

- The readme.txt file provides a summary of where to find files in a plain text format.
- Copyright.txt and Acknowledgements.txt provide copyright information and legal notices for third party and open-source software used in Delinea Server Suite.
- Delinea-end-user-license-agreement.txt provides the text of the license agreement displayed during installation.
- autorun.inf controls the autorun program, autorun.exe, on Windows computers.

The following are sub-folders that are organized to provide you access to different software components in the Delinea Server Suite.

- Agent folder

  This folder contains the installer packages for installing Server Suite Agent for Windows on Windows computers.

- Common folder

  This folder contains the installer packages for common components necessary for all Delinea products on Windows computers.

- DirectAudit folder

  This folder contains the installer packages for Delinea Audit & Monitoring Service on Windows computers.

- DirectManage folder

  This folder contains the installer packages for Delinea Authentication Service and Delinea Privilege Elevation Service on Windows computers.

- LicensingService Folder

This folder contains the installer packages for Delinea Licensing Service utilities on Windows computers.

- Resources Folder

  This folder contains resources for internal use for the media. It can be safely ignored.

Server Suite Agents for UNIX/Linux

This image contains a zipped bundle of files for Server Suite agent on each supported UNIX, or Linux platform and an adcheck utility for each supported platform.

You may find the appropriate bundle for an OS platform based on the following table:

delinea-server-suite-<release number>-aix7.1-ppc.tgz	IBM AIX, IBM VIOS
delinea-server-suite-<release number>-cos-x86_64.tgz	Flatcar, RHCOS
delinea-server-suite-<release number>-deb9-arm64.tgz	Ubuntu
delinea-server-suite-<release number>-deb9-ppc64el.tgz	Ubuntu
delinea-server-suite-<release number>-deb9-x86_64.tgz	Debian, Ubuntu
delinea-server-suite-<release number>-hp11.31-ia64.tgz	HPUX
delinea-server-suite-<release number>-hp11.31-pa.tgz	HPUX
delinea-directcontrol-<release number>-mac10.15.tgz	MAC (Intel, M1)
delinea-server-suite-<release number>-rhel6-ppc64.tgz	RHEL
delinea-server-suite-<release number>-rhel6-x86_64.tgz	Amazon Linux, CentOS, Fedora, Oracle Linux, RHEL, AlmaLinux, Rocky Linux
delinea-server-suite-<release number>-rhel7-aarch64.tgz	Amazon Linux, CentOS, Oracle Linux, RHEL
delinea-server-suite-<release number>-rhel7-ppc64le.tgz	RHEL
delinea-server-suite-<release number>-sol10-sparc.tgz	Oracle Solaris
delinea-server-suite-<release number>-sol10-x86.tgz	Oracle Solaris
delinea-server-suite-<release number>-sol11-i386.tgz	Oracle Solaris (IPS package)
delinea-server-suite-<release number>-sol11-sparc.tgz	Oracle Solaris (IPS package)
delinea-server-suite-<release number>-suse12-aarch64.tgz	SUSE
delinea-server-suite-<release number>-suse12-ppc64le.tgz	SUSE
delinea-server-suite-<release number>-suse12-x86_64.tgz	SUSE

Notes:

- The OS version number specified in the bundle name indicates the minimum OS version that it supports.
- You should also choose the appropriate bundle for the specific architecture as indicated in the bundle name.
- Inside each bundle, it contains packages of associated products supported on that platform. The naming convention follows the above bundle names except that the prefix of a package reflects the product it serves. The following are the possible package prefixes and the corresponding product

names:

CentrifyDA	Delinea DirectAudit package
CentrifyDC	Delinea DirectControl package
CentrifyDC-cifsidmap	Delinea for CIFS ID mapping package
CentrifyDC-curl	Required component of Delinea DirectControl package
CentrifyDC-ldapproxy	Delinea OpenLDAP Proxy package
CentrifyDC-nis	Delinea Network Information Service and Delinea NIS Server package
CentrifyDC-openldap	Required component of Delinea DirectControl package
CentrifyDC-openssh	Delinea OpenSSH package
CentrifyDC-openssl	Required component of Delinea DirectControl package

- Before installation, please review the *Upgrade and Compatibility Guide* and run the adcheck utility to make sure the environment is ready, especially if you are using native package manager to install.

Go to Supported Versions.

Newly Added Supported Platforms

Support is added for the following operating system platforms in this release:

- AlmaLinux 8.5
- Flatcar
- Red Hat Enterprise Linux CoreOS (RHCOS)
- Rocky Linux 8.5

Supported UNIX/Linux Platforms

AlmaLinux 8.5	x86_64	Yes	Yes	Yes	
Alpine Linux 3.13, 3.14	X86_64	No	Yes	Yes	
Amazon Linux 2 LTS	aarch64	No	Yes	Yes	
Amazon Linux 2 LTS	x86_64	No	Yes	Yes	
CentOS 7.4-7.9, 8.0-8.5	aarch64	No	Yes	Yes	
CentOS 6.0-6.10, 7.0-7.9, 8.0-8.5	x86_64	Yes	Yes	Yes	
Debian 9.0-9.13, 10.0-10.11, 11	x86_64	Yes	Yes	Yes	

Flatcar	x86_64	No	Yes	Yes	
HP-UX 11.31 (Trusted and Untrusted)	Itanium	No	Yes	Yes	
HP-UX 11.31 (Trusted and Untrusted)	PA-RISC	No	Yes	Yes	
IBM AIX 7.1 TL1+, 7.2	ppc	No	Yes	Yes	Note 4
IBM Virtual I/O Server 3.x	ppc	No	Yes	Yes	
MacOS 11.0-11.6, 12	M1	No	Yes	No	Note 3
MacOS 10.15, 11.0-11.6, 12	x86_64	No	Yes	No	Note 3
Oracle Linux 7.4-7.9, 8.0-8.5	aarch64	No	Yes	Yes	
Oracle Linux 6.0-6.10, 7.0-7.9, 8.0-8.5	x86_64	Yes	Yes	Yes	
Oracle Solaris 10 u8+, 11.0-11.4	SPARC	No	Yes	Yes	Note 2
Oracle Solaris 10 u8+, 11.0-11.4	x86_64	No	Yes	Yes	Note 2
Red Hat Enterprise Linux 7.4-7.9, 8.0-8.5	aarch64	No	Yes	Yes	
Red Hat Enterprise Linux 6.0-6.10, 7.0-7.9	ppc64	Yes	Yes	Yes	
Red Hat Enterprise Linux 7.1-7.9, 8.0-8.5	ppc64le	Yes	Yes	Yes	
Red Hat Enterprise Linux 8.0	S390	No	Yes	No	
Red Hat Enterprise Linux 6.0-6.10, 7.0-7.9, 8.0-8.5	x86_64	Yes	Yes	Yes	
Red Hat Enterprise Linux CoreOS (RHCOS)	x86_64	Yes	Yes	Yes	Note 1
Red Hat Fedora Linux 34, 35	x86_64	Yes	Yes	Yes	
Rocky Linux 8.5	x86_64	Yes	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	aarch64	No	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	ppc64le	Yes	Yes	Yes	
SUSE Enterprise Linux 12 SP4	S390	No	Yes	Yes	
SUSE Enterprise Linux 12 SP3+, 15	x86_64	Yes	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 21.04, 21.10	arm64	No	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 21.04, 21.10	ppc64el	Yes	Yes	Yes	
Ubuntu Linux 18.04, 20.4, 21.04, 21.10	x86_64	Yes	Yes	Yes	

Note 1: Please refer to the Unexpected Link Text for features supported on this platform.

Note 2: Starting with Release 2020, we require the OS patch level update 8 or above on Solaris 10.

Note 3: Delinea OpenSSH is not supported on this platform.

Note 4: Starting with Release 2021.1, we require the TL1 or above on AIX 7.1.

Additional Information

You should follow the OS vendors' recommendation to update the necessary patches. Here are the minimum patch requirements for the specific UNIX platforms (Ref: CS-45562):

1. HPUX 11.31
    1. PHNE_40225 - Cumulative Console and BSD Pty Patch (it is required for DirectAudit package)
2. Solaris 10 x86_64
    1. 119255-66
    2. 127128-11
    3. 141445-09
    4. 142910-17
3. Solaris 10 SPARC
    1. 119254-66
    2. 120011-14
    3. 127127-11
    4. 142909-17

Supported Windows Platforms

The following 64-bit Windows platforms are supported on Delinea Server Suite (Ref: CS-49379):

- Windows 10 LTSB/LTSC (Note 1)
- Windows 11 LTSB/LTSC
- Windows Server 2012
- Windows Server 2012R2
- Windows Server 2016
- Windows Server 2019 LTSC
- Windows Server 2022 LTSC
- Windows Server 2012 Core (Note 2)
- Windows Server 2012 Minimum Server Interface (Note 2)
- Windows Server 2012R2 Core (Note 2)
- Windows Server 2012R2 Minimum Server Interface (Note 2)

Note:

1. We support Windows 10 Long Term Servicing Channel (LTSC), or previously called Long Term Servicing Branch (LTSB), editions based on Microsoft's lifecycle fact sheet https://docs.microsoft.com/en-us/lifecycle/faq/windows and https://docs.microsoft.com/en-us/windows/release-health/release-information
2. Only the Privilege Elevation Service component of Server Suite Agent for Windows supports these platforms (Core and Minimum Server Interface)
3. Support for all 32-bit Windows platform was terminated in 2015 (Server Suite 2015.1)

   Also note that Server Suite require specific versions of .NET to work. Please refer to the following table for the requirement (Ref: CS-49381):

Server Suite 2022	April 2022	4.8	--*
Server Suite 2021.1	December 2021	4.8	--*
Server Suite 2021	July 2021	4.8	--*
Infrastructure Services 2020.1	December 2020	4.6.2	--*

Infrastructure Services 2020	September 2020	4.6.2	--*
Infrastructure Services 19.9	December 2019	4.6.2	--*
Infrastructure Services 19.6	August 2019	4.6.2	--*
Infrastructure Services 18.11	December 2018	4.6.2	4.6.2
Infrastructure Services 18.8	August 2018	4.6.2	4.6.2
Infrastructure Services 2018	April 2018	4.6.2	4.6.2
Infrastructure Services 2017.3	December 2017	4.5.2	4.5.2
Infrastructure Services 2017.2	September 2017	4.5.2	4.5.2
Server Suite 2017.1	May 2017	4.5	4.5.2
Server Suite 2017	February 2017	4.5	4.5.2
Server Suite 2016.1	May 2016	4.5	4.5.2
Server Suite 2016	December 2016	4.5	4.5.2

Note: We no longer bundle .NET in our installation media any more starting Release 19.6. (Ref: CS-47940)

This is the last release to support the following operating system platforms:

- Ubuntu 21.04, 21.10
- Fedora 34

Delinea has established product security policies documented in this web page. You may also find the details of all the published security advisories there.

For component specific security fixes in this release, you may find them in the corresponding component release-notes.html files. Please refer to Section 7 for a description of individual release notes.

See Component Version Table.

- For Access Manager, DirectControl agent and Delinea OpenSSH, see the Authentication Service and Privilege Elevation Service Release Notes.
- For Audit Manager and DirectAudit agent, see the Audit & Monitoring Service Release Notes.
- For Agent for Windows, see the Agent for Windows Release Notes.
- For Server Suite PuTTY, see the Server Suite PuTTY Release Notes

You can get all the supported releases from the download center in Delinea support web site.

- All the ISO, ZIP, and TGZ files are associated with the MD5 checksum.
- All RPM and DEB packages as well as YUM and APT repositories are also protected by the GPG signature. You can find the GPG public key in the

download center.

Component specific bug fixes in this release can be found in the corresponding component release notes files. Please refer to Release Notes for Server Suite Components for a description of individual release notes.

Component specific known issues/limitations can be found in the corresponding component release notes files. Please refer to Release Notes for Server Suite Components for a description of individual release notes.

For the most up to date list of known issues, please login to the Customer Support Portal at https://www.delinea.com/support and refer to Knowledge Base articles for any known issues with the release.

In addition to the documentation provided with this package, you can find the answers to common questions and information about any general or platform-specific known limitations as well as tips and suggestions from the Delinea Knowledge Base.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

This section lists the recent release notes for Server Suite for Mac.

- Release 2022
- 2020 Release Notes -- Adbindproxy (Samba)

DirectControl for Samba is a proxy agent package that seamlessly integrates the Server Suite Agent for *NIX with open source Samba (referred to as stock Samba in this document), enabling the two products to share Active Directory user and group membership and to agree upon Unix identity attributes for Active Directory users. It is a proxy that passes identity management requests from Samba to the Server Suite Agent for *NIX.

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement.

The DirectControl for Samba bundle package contains the following resources:

- DirectControl for Samba software package

  (for example, CentrifyDC-adbindproxy-<version#>-<OS>.<architecture>.rpm, or similar platform specific package file)

The DirectControl for Samba bundle package is available on the following OS/platforms in this release:

- IBM AIX on PPC
- Oracle Solaris on SPARC
- Oracle Solaris on x86_64
- Ubuntu on x86_64
- Red Hat Enterprise Linux on PPC
- Red Hat Enterprise Linux on PPC64LE
- Red Hat Enterprise Linux on x86_64
- SUSE Linux Enterprise Server on x86_64

  This DirectControl for Samba release supports stock Samba version 4.14 to 4.16. You are strongly advised to apply the latest security patches from Samba first before deploying DirectControl for Samba.

  For the OS versions that a DirectControl for Samba bundle package supports, please refer to the supported OS versions of the matching DirectControl Agent for *NIX package of the corresponding Server Suite release. Similarly, DirectControl for Samba also follows DirectControl Agent for *NIX's schedule for End-of-Support platforms and hence please refer to the announcements there.

No new features in this release.

The following sections describe common known issues or limitations associated with this DirectControl for Samba release.

- Limitations with stock Samba

  In previous releases of DirectControl for Samba, we modified the following in stock Samba for interoperability. Using stock Samba instead of Centrify Samba, you may see related issues.

  - Default Kerberos keytab location, KEYTAB_DEFAULT, from /etc/krb5.keytab to /etc/krb5/krb5.keytab on Solaris (SAMBA-890).

- Default Kerberos cache location, CCNAME, from /tmp/krb5cc_% to /var/krb5/security/creds/krb5cc_%" on AIX (SAMBA-892).

- Limitations with RHEL 7.2 PPC (SAMBA-965)

  If you are using 64bit Samba on a RHEL 7.2 PPC machine, you may have problem with adclient failed to use the 64bit tdb library come with 64 bit Samba. The symptom can be shown in the error message while trying to access samba server - "session setup failed: NT_STATUS_CANT_ACCESS_DOMAIN_INFO".

  You need to install a 32bit tdb library, such as libtdb-1.3.6-2.el7.ppc.rpm in rhel-server-7.2-ppc64-dvd.iso, for adclient to work with, and you need to tell adclient where to get this library by adding a parameter "samba.libtdb.path: /usr/lib/libtdb.so.1" into centrifydc.conf, assuming the path to libtdb is /usr/lib/libtdb.so.1.

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

This section lists the recent release notes for Server Suite for Mac.

- Server Suite for Mac 2022.1
- Server Suite for Mac 2022

You can find release notes and documentation related to previous releases at Previous Releases.

Server Suite for Mac provides Active Directory-based authentication, single sign-on, and group policy support for the macOS platform.

Server Suite for Mac is a part of Delinea software and is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962.

What's Included in this Release

- CentrifyDC-5.9.0-mac10.15.dmg: A Mac disk image for macOS 12.x, 11.x, and 10.15 containing the following:
  - AD Check.app: Graphical application to perform environment checks before installing Server Suite on macOS 12.x, 11.x, and 10.15
  - CentrifyDC-5.9.0-x86_64.pkg: Graphical installer of Server Suite for Macs (valid on both Intel and Apple Silicon) on macOS 12.x, 11.x, and 10.15

Supported Platforms and System Requirements

The Server Suite for Mac in the applicable package can be installed on the following versions of the macOS operating system:

- macOS 12.x on both Intel and Apple Silicon
- macOS 11.x on both Intel and Apple Silicon
- macOS 10.15.x on Intel

Installing on macOS 12 Monterey

If you are running the current release of Server Suite, you MUST UPGRADE Server Suite BEFORE upgrading your Mac to macOS 12 Monterey.

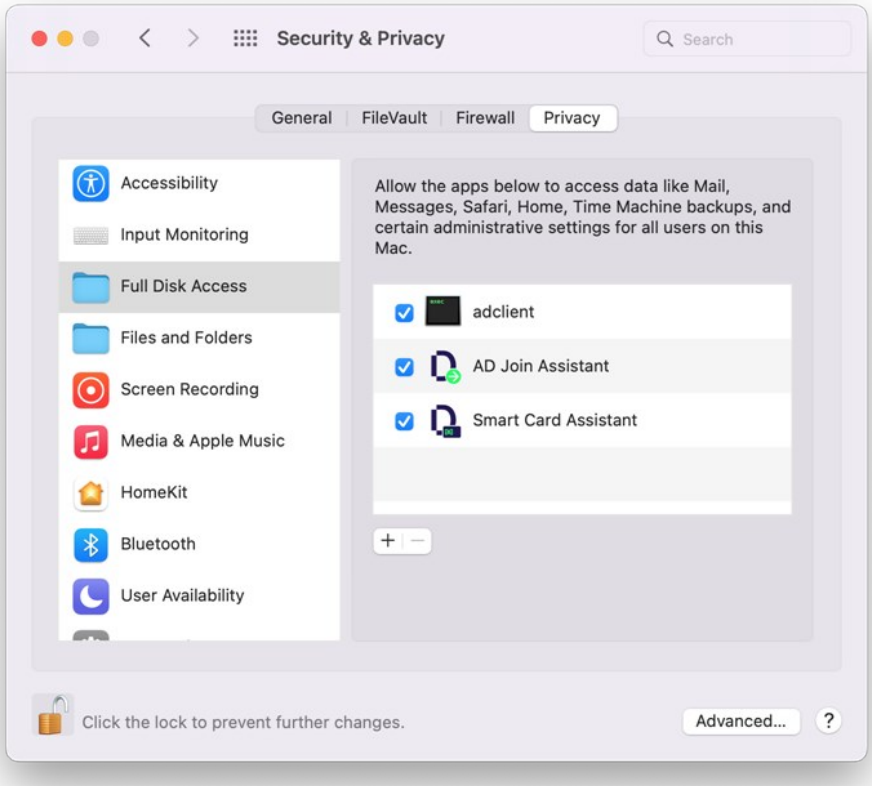
Follow these steps:

1. Download the Server Suite package for macOS.
2. Upgrade Server Suite for macOS using the package you downloaded.
3. Upgrade to macOS 12.

Setting Full Disk Access for the DirectControl Agent

Due to a limitation of macOS 11.x and macOS 12.x, "Full Disk Access" is required for the DirectControl Agent for Mac. You can configure this yourself if you're an administrator on the computer, or you can set it by way of your MDM (Mobile Device Management) provider.

1. To configure full disk access as an administrator:
    1. Log in to the Mac as an admin user.
    2. Open **System Preferences**.
    3. Click **Security & Privacy**.
    4. Click **Privacy**.
    5. Click the **Lock** button to input password or use TouchID to unlock.
    6. Scroll down a little bit on the left list, find and select **Full Disk Access**.
    7. Click the **Plus** button.
    8. Press and hold these three keys together: shift + command + G.
    9. Input the path /usr/local/sbin/adclient and click **GO**, then click **Open** to add it.
    10. Repeat step 7 and 8, then input the path /Applications/Utilities/Centrify/AD Join Assistant.app and click **GO**, then click **Open** to add it.
    11. Repeat step 7 and 8, then input the path /Applications/Utilities/Centrify/Smart Card Assistant.app and click **GO**, then click **Open** to add it.
    12. Click the **Lock** button again to lock.



2. Configure full disk access through your MDM provider. Contact your MDM provider for more information.

   Your MDM provider will need the following information:

```
% codesign -dv /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
Identifier=adclient
...
% codesign -dr - /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
designated => identifier adclient and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"

% codesign -dv /Applications/Utilities/Centrify/AD\ Join\ Assistant.app
Executable=/Applications/Utilities/Centrify/AD Join Assistant.app/Contents/MacOS/AD Join Assistant
Identifier=com.centrify.cdc.centrifyjoinassistant
...
% codesign -dr - /Applications/Utilities/Centrify/AD\ Join\ Assistant.app

Executable=/Applications/Utilities/Centrify/AD Join Assistant.app/Contents/MacOS/AD Join Assistant
designated => identifier "com.centrify.cdc.centrifyjoinassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.   6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"

% codesign -dv /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
Identifier=com.centrify.cdc.smartcardassistant
...
% codesign -dr - /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
designated => identifier "com.centrify.cdc.smartcardassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.   6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"
```

Feature Changes and Notable Fixes in this Release

- The application "Centrify Join Assistant" is now called the "AD Join Assistant".

Known macOS Issues

- As of macOS Big Sur, Apple no longer permits to silently install configuration profiles. It affects the following group policies and they will not work on macOS Big Sur:
    1. Group policy "Install MobileConfig Profiles"
    2. Group policy "Enable Profile Custom Settings"
    3. Group policy "Require password to wake this computer from sleep or screen saver"
    4. Group policy "Enable Machine Ethernet Profile"
    5. Group policy "Enable Machine Wi-Fi Profile"
    6. Group policy "Enable User Ethernet Profile"
    7. Group policy "Enable User Wi-Fi Profile"
- When upgrading Mac from macOS 10.14 or lower to macOS 10.15 or higher, you must install the new agent version. You don't need to leave the domain or uninstall the old CentrifyDC agent.
- A network user doesn't work on macOS 10.15 and higher. We suggest using a mobile user or a general Active Directory user instead.
- When a mobile user logs in for the first time on macOS Big Sur and higher, in some cases they cannot set up Touch ID with their fingerprints. They just need to re-login in order for Touch ID to work.

Apple Support has provided the following resolutions:

* [Reset the SMC of Mac](https://support.apple.com/en-us/HT201295)
* [Reset NVRAM or PRAM on Mac](https://support.apple.com/en-us/HT204063)

Notice of Termination of Support

Server Suite has discontinued support for Mac OS 10.14.x, 10.13.x, 10.12.x, and 10.11.x starting with this 2022 release of Server Suite for Mac.

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

Server Suite for Mac provides Active Directory-based authentication, single sign-on, and group policy support for the macOS platform.

Server Suite for Mac is a part of Delinea software and is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962.

What's Included in this Release

- CentrifyDC-5.9.1-mac10.15.dmg: A macOS disk image for macOS 13.x, 12.x, 11.x, and 10.15 that contains contains the following:
  - AD Check.app: A graphical application to perform environment checks before installing Server Suite on macOS 13.x, 12.x, 11.x, and 10.15
  - CentrifyDC-5.9.1.pkg: A graphical installer of Server Suite for Mac (valid on both Intel and Apple Silicon) on macOS 13.x, 12.x, 11.x, and 10.15

Supported Platforms and System Requirements

The Server Suite for Mac in the applicable package can be installed on the following versions of the macOS operating system:

- macOS 13.x on both Intel and Apple Silicon
- macOS 12.x on both Intel and Apple Silicon
- macOS 11.x on both Intel and Apple Silicon
- macOS 10.15.x on Intel

Installing on macOS 13 Ventura

If you are running the current release of Server Suite, you MUST UPGRADE Server Suite BEFORE upgrading your Mac to OS 13 Ventura.

Follow these steps:

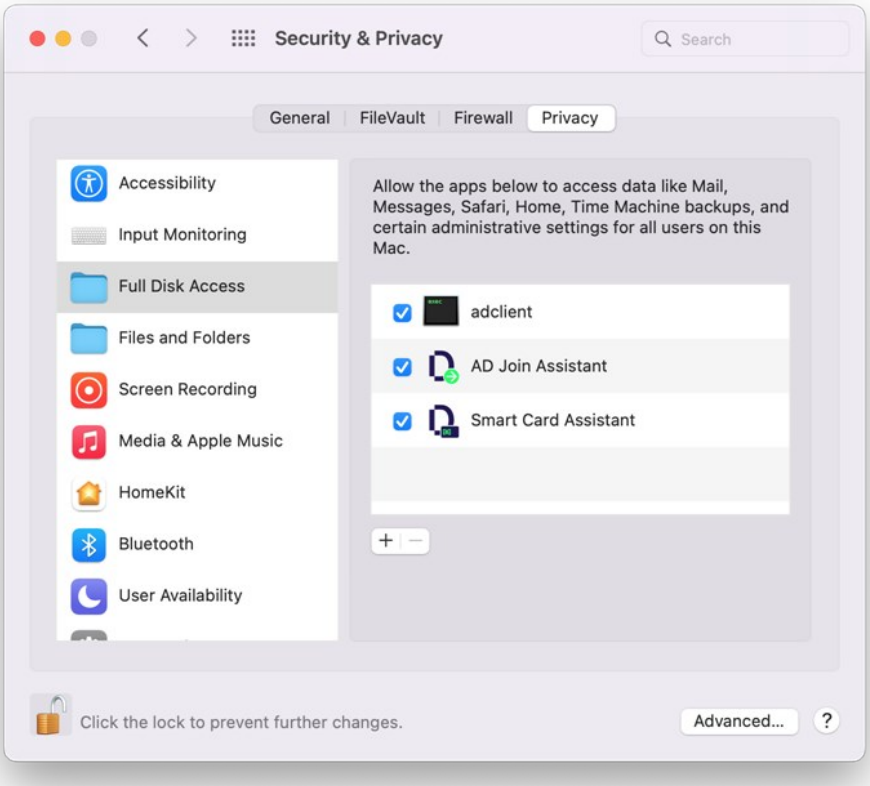
1. Download the Server Suite package for macOS.
2. Upgrade Server Suite for macOS using the package you downloaded.
3. Upgrade to macOS 13.

Setting Full Disk Access for the DirectControl Agent

Due to a limitation of macOS 11.x, macOS 12.x, and macOS 13.x, "Full Disk Access" is required for the DirectControl Agent for Mac. You can configure this yourself if you're an administrator on the computer, or you can set it by way of your MDM (Mobile Device Management) provider.

1. To configure full disk access as an administrator:
    1. Log in to the Mac as an admin user.
    2. Open **System Preferences**.
    3. Click **Security & Privacy**.
    4. Click **Privacy**.
    5. Click the **Lock** button to input password or use TouchID to unlock.
    6. Scroll down a little bit on the left list, find and select **Full Disk Access**.
    7. Click the **Plus** button.
    8. Press and hold these three keys together: shift + command + G.
    9. Input the path /usr/local/sbin/adclient and click **GO**, then click **Open** to add it.
    10. Repeat step 7 and 8, then input the path /Applications/Utilities/Centrify/AD Join Assistant.app and click **GO**, then click **Open** to add it.
    11. Repeat step 7 and 8, then input the path /Applications/Utilities/Centrify/Smart Card Assistant.app and click **GO**, then click **Open** to add it.

12. Click the **Lock** button again to lock.



2. Configure full disk access through your MDM provider. Contact your MDM provider for more information.

   Your MDM provider will need the following information:

```
% codesign -dv /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
Identifier=adclient
...
% codesign -dr - /usr/local/sbin/adclient
Executable=/usr/local/sbin/adclient
designated => identifier adclient and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
certificate leaf[subject.OU] = "64CT837G5Z"

% codesign -dv /Applications/Utilities/Centrify/AD\ Join\ Assistant.app
Executable=/Applications/Utilities/Centrify/AD Join Assistant.app/Contents/MacOS/AD Join Assistant
Identifier=com.centrify.cdc.centrifyjoinassistant
...
% codesign -dr - /Applications/Utilities/Centrify/AD\ Join\ Assistant.app

Executable=/Applications/Utilities/Centrify/AD Join Assistant.app/Contents/MacOS/AD Join Assistant
designated => identifier "com.centrify.cdc.centrifyjoinassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.   6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"

% codesign -dv /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
Identifier=com.centrify.cdc.smartcardassistant
...
% codesign -dr - /Applications/Utilities/Centrify/Smart\ Card\ Assistant.app
Executable=/Applications/Utilities/Centrify/Smart Card Assistant.app/Contents/MacOS/SCTool
designated => identifier "com.centrify.cdc.smartcardassistant" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.   6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "64CT837G5Z"
```

Feature Changes and Notable Fixes in this Release (Release 2022.1 Component Update)

- This release supports macOS 13 "Ventura". (Ref: 486761)
- DirectControl now natively supports Apple Silicon. Rosetta 2 is no longer needed. (Ref: 486761)

Known macOS Issues

- As of macOS Big Sur, Apple no longer permits to silently install configuration profiles. It affects the following group policies and they will not work on macOS Big Sur:
    1. Group policy "Install MobileConfig Profiles"
    2. Group policy "Enable Profile Custom Settings"
    3. Group policy "Require password to wake this computer from sleep or screen saver"
    4. Group policy "Enable Machine Ethernet Profile"
    5. Group policy "Enable Machine Wi-Fi Profile"
    6. Group policy "Enable User Ethernet Profile"
    7. Group policy "Enable User Wi-Fi Profile"
- When upgrading Mac from macOS 10.14 or lower to macOS 10.15 or higher, you must install the new agent version. You don't need to leave the domain or uninstall the old CentrifyDC agent.
- A network user doesn't work on macOS 10.15 and higher. We suggest using a mobile user or a general Active Directory user instead.
- When a mobile user logs in for the first time on macOS Big Sur and higher, in some cases they cannot set up Touch ID with their fingerprints. They just need to re-login in order for Touch ID to work.

Apple Support has provided the following resolutions:

* [Reset the SMC of Mac](https://support.apple.com/en-us/HT201295)
* [Reset NVRAM or PRAM on Mac](https://support.apple.com/en-us/HT204063)

Notice of Termination of Support

Server Suite has discontinued support for Mac OS 10.14.x, 10.13.x, 10.12.x, and 10.11.x starting with the 2022 release of Server Suite for Mac.

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

This section lists the recent release notes for Server Suite PuTTY.

About this Release

PuTTY is a popular open-source client on Windows and UNIX-based computers that provides access to remote machines using well-known network protocols such as Telnet, SSH, rlogin and raw TCP. However, it does not support Kerberos authentication.

Delinea has enhanced PuTTY so that user authentication can be accomplished using Kerberos before establishing a remote connection. The Delinea-enhanced version of PuTTY takes advantage of the Kerberos environment that the Delinea DirectControl agent sets up automatically on managed UNIX and Linux computers. By using the Delinea PuTTY client to access Delinea-managed computer, you gain the benefits of centralized authentication and password policy enforcement using a secure and well-established authentication infrastructure.

Delinea PuTTY is currently integrated with open-source PuTTY version 0.75.

The Delinea Server Suite and PuTTY release notes and documents are available online at https://docs.delinea.com/.

Feature Changes

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Fixed Issues

Known Issues

- puttytel does not support Kerberos authentication.
- If you specify Alternate Kerberos credentials on the SSH > Kerberos properties page, you will always be prompted for a password. This will happen even if you choose to remember the password when first prompted for it.

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.

About this Release

PuTTY is a popular open-source client on Windows and UNIX-based computers that provides access to remote machines using well-known network protocols such as Telnet, SSH, rlogin and raw TCP. However, it does not support Kerberos authentication.

Delinea has enhanced PuTTY so that user authentication can be accomplished using Kerberos before establishing a remote connection. The Delinea-enhanced version of PuTTY takes advantage of the Kerberos environment that the Delinea DirectControl agent sets up automatically on managed UNIX and Linux computers. By using the Delinea PuTTY client to access Delinea-managed computer, you gain the benefits of centralized authentication and password policy enforcement using a secure and well-established authentication infrastructure.

Delinea PuTTY is currently integrated with open-source PuTTY version 0.75.

Delinea software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,378,391 and 9,442,962. (Ref: CS-44575)

Feature Changes

Server Suite and its component services have been changed to use the new Delinea name and logo.

For more information about Delinea, see Delinea Announcement

Fixed Issues

- Fixed a crash in Putty during startup or when configuring or loading the settings under Connection->SSH->Kerberos. (Ref: 425942)

Known Issues

- puttytel does not support Kerberos authentication.
- If you specify Alternate Kerberos credentials on the SSH > Kerberos properties page, you will always be prompted for a password. This will happen even if you choose to remember the password when first prompted for it.

Additional Information and Support

In addition to the documentation provided with this package, see the Delinea Knowledge Base for answers to common questions and other information (including any general or platform-specific known limitations), tips, or suggestions. You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone.

The Delinea Resources web site provides access to a wide range of information including analyst report, best practice brief, case study, datasheet, ebook, white papers, etc., that may help you optimize your use of Delinea products. For more information, see the Delinea Resources web site.

You can also contact Delinea Support directly with your questions through the Delinea Web site, by email, or by telephone. To contact Delinea Support or to get help with installing or using this software, send email to support@delinea.com or call 1-202-991-0540. For information about purchasing or evaluating Delinea products, send email to info@delinea.com.