Administrator's Guide for Linux and UNIX

September 2020 (release 2020)

Centrify Corporation

Legal Notice

Contents

Troubleshooting authentication and authorization 260

Using Centrify commands for administrative tasks 286

About Centrify Management Services for Linux and Unix

The *Administrator's Guide for Linux and UNIX* describes how to use Centrify software to manage user and group profiles, role-based access rights, and delegated administrative activity for Linux and UNIX computers. This guide focuses exclusively on the management of identity attributes, rights, roles, role assignments, and privileges that apply to Linux and UNIX computers. If you manage a heterogeneous environment that includes Linux, UNIX, Mac OS X, and Windows computers, you should check for additional information in the other guides that make up the Centrify documentation set.

Intended audience

The *Administrator's Guide for Linux and UNIX* is intended for administrators who are responsible for managing user access to servers, workstations, enterprise applications, and network resources. This guide focuses on using Centrify Access Manager and related software components to administer Centrify-managed UNIX and Linux computers, and on deploying the same authentication and policy services deployed you use for Windows computers. You can perform the same administrative tasks described in this guide using a variety of other tools, but you should know how to perform common administrative tasks on the operating systems you support.

You should note that this guide does not cover deployment planning or installation details. For complete information about planning and installing Centrify software, see the *Planning and Deployment Guide*.

Documentation conventions

The following conventions are used in Centrify documentation:

- `Fixed-width` font is used for sample code, program names, program output, file names, and commands that you type at the command line. When *italicized*, this font indicates variables. Square brackets (`[ ]`) indicate optional command-line arguments.
- **Bold** text is used to emphasize commands or key command results; buttons or user interface text; and new terms.
- *Italics* are used for book titles and to emphasize specific words or terms. In fixed-width font, italics indicate variable values.
- Standalone software packages include version and architecture information in the file name. Full file names are not documented in this guide. For complete file names for the software packages you want to install, see the distribution media.
- For simplicity, UNIX is used to refer to all supported versions of the UNIX and Linux operating systems. Some parameters can also be used on Mac OS X computers.

Finding more information about Centrify products

Centrify provides extensive documentation targeted for specific audiences, functional roles, or topics of interest. If you want to learn more about Centrify and Centrify products and features, start by visiting the Centrify website. From the Centrify website, you can download data sheets and evaluation software, view video demonstrations and technical presentations about Centrify products, and get the latest news about upcoming events and webinars.

For access to documentation for all Centrify products and services, visit the Centrify documentation portal at docs.centrify.com. From the Centrify documentation portal, you can always view or download the most up-to-date version of this guide and all other product documentation.

For details about supported platforms, please consult the release notes.

For the most up to date list of known issues, please login to the Customer Support Portal at http://www.centrify.com/support and refer to Knowledge Base articles for any known issues with the release.

Product names

Over the years we've made some changes to some of our product offerings and features and some of these previous product names still exist in some areas. Our current product offerings include the following services:

Current Overall Product Name	Current Services Available
Centrify Identity-Centric PAM	Privileged Access Service
	Gateway Session Audit and Monitoring
	Authentication Service
	Privilege Elevation Service
	Audit and Monitoring Service
	Privilege Threat Analytics Service

Whether you're a long-time or new customer, here are some quick summaries of which features belong to which current product offerings:

Previous Product Offering	Previous Product Offering	Description	Current Product Offering
	Centrify Privileged Service (CPS)		Privileged Access Service
DirectControl (DC)			Authentication Service
DirectAuthorize (DZ or DZwin)			Privilege Elevation Service
DirectAudit (DA)			Audit and Monitoring Service
	Infrastructure Services		Privileged Access Service, Authentication Service, Privilege Elevation Service, Audit and Monitoring Service, and Privilege Threat Analytics Service
DirectManage (DM)	Management Services	Consoles that are used by all 3 services: Authentication Service, Privilege Elevation Service, and Audit and Monitoring Service	

Previous Product Offering	Previous Product Offering	Description	Current Product Offering
DirectSecure (DS)	Isolation and Encryption Service		Still supported but no longer being developed or updated
	User Analytics Service		Privilege Threat Analytics Service
Deployment Manager		Deployment Manager provided a centralized console for discovering, analyzing, and managing remote computers. This feature is no longer included starting with Infrastructure Services release 19.6.	

Depending on when you purchased a Centrify product offering, you may have purchased one of the following product bundles:

Previous Product Bundle	Previous Product Bundle	Current Product Bundle	Services Included	Description
		Centrify Identity-Centric PAM Core Edition	Privileged Access Service and Gateway Session Audit and Monitoring	
Centrify Server Suite Standard Edition			Authentication Service and Privilege Elevation Service	
	Centrify Infrastructure Services Standard Edition	Centrify Identity-Centric PAM Standard Edition	Privileged Access Service, Authentication Service, and Privilege Elevation Service	
Centrify Server Suite Enterprise Edition			Authentication Service, Privilege Elevation Service, and Audit and Monitoring Service	

Previous Product Bundle	Previous Product Bundle	Current Product Bundle	Services Included	Description
	Centrify Infrastructure Services Enterprise Edition	Centrify Identity-Centric PAM Enterprise Edition	Privileged Access Service, Authentication Service, Privilege Elevation Service, Audit and Monitoring Service (includes Gateway Session Audit and Monitoring)	
Centrify Server Suite Platinum Edition				Discontinued bundle that included DirectControl, DirectAuthorize, DirectManage, DirectAudit, and DirectSecure

Contacting Centrify

You can contact Centrify by visiting our website, www.centrify.com. On the website, you can find information about Centrify office locations worldwide, email and phone numbers for contacting Centrify sales, and links for following Centrify on social media. If you have questions or comments, we look forward to hearing from you.

Getting additional support

If you have a Centrify account, click Support on the Centrify website to log on and access the Centrify Technical Support Portal. From the support portal, you can search knowledge base articles, open and view support cases, download software, and access other resources.

To connect with other Centrify users, ask questions, or share information, visit the Centrify Community website to check in on customer forums, read the latest blog posts, view how-to videos, or exchange ideas with members of the community.

Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service for Linux and UNIX

Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service is an IT management solution that provides key services for managing user and group profiles, role-based access rights, elevated privileges for administrative activity, and auditing-based regulatory compliance. These services can be used together or independently, depending on the requirements of your organization. The topics in this section introduce the key Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service that enable you to centrally manage Linux and UNIX computers. It includes an overview of how Centrify enables your organization to manage identity attributes, role-based access rights, and administrative activity through an integrated set of services.

Why securing access is crucial

For most organizations, it is critical to control access to computer and application resources to prevent disruptions of service, data tampering, or security breaches. For many organizations, it is also critical to monitor and report on user activity to ensure regulatory compliance with government or industry standards. However, managing who has access to sensitive data, core business services, and the computers and applications that perform vital functions is especially difficult in data centers that include a mix of virtual and physical computers running different operating systems and platform versions.

Why managing user account information might be a problem

In a cross-platform environment, you are likely to have multiple identity stores that might have overlapping or conflicting information about the user population. You might also have several different authentication methods—with varying degrees of security—that you are required to manage. For example, in a typical environment with a mix of Linux and UNIX computers, you might have to maintain any combination of the following authentication methods:

- Local configuration files on individual UNIX servers and workstations to identify local users and groups.
- NIS or NIS+ servers and maps to store account and network information for groups of UNIX servers and workstations.
- Kerberos realms and a Key Distribution Center to provide authentication for some users and services.
- Lightweight Directory Access Protocol services to support LDAP queries and responses.

Managing all of these services separately can be costly and inefficient. In addition, users who have access to more than one application or computer platform often have to remember multiple login accounts with conflicting user name or password policy requirements. Individual applications might also require the use of a specific authentication method. For example, a database application or a web service might require users to have a database- or application-specific account.

If you have an environment where user and group account information is stored in multiple locations rather than in a single repository, it is likely that you have overlapping, conflicting, or out-of-date information about who should have access to the computers in your organization. You might also be using less secure authentication and authorization services than required, if you are relying on local configuration files or NIS servers and maps. For example, if you are in an organization that is subject to regulatory compliance, an audit might require you to improve the security of the authentication and authorization services you use.

Why managing access and privileges might be a problem

Most organizations require some groups of users to be allowed to use administrative accounts and passwords. For example, you might want to grant these permissions to allow some users to log on to computers that host administrative applications or data center services, but restrict access so that users can only log on when appropriate.

In many cases, the primary way you secure access to computers is by granting a limited number of users or groups `root` administrative privileges or configuring `sudoers` rights locally. These common practices leave computers vulnerable to insider threats and present a security risk that might be exploited by an external attack. As common as it is, granting administrative access rights is likely to violate the principal of least privilege, which is intended to minimize your exposure to these types of risks.

In other cases, users who need administrative privileges to perform specific tasks might use a shared administrator and service account password. However, shared passwords reduce accountability, leave computers vulnerable to insider threats, and are also often flagged by auditors as a security issue. If you are in an industry that has compliance requirements, shared passwords might present a significant business risk.

How Centrify can reduce security risks

To reduce the overhead of managing account information and access rights across your organization, Centrify provides the following key features:

Secure authentication and identity management

Centrify enables you to define and manage the identity attributes in user profiles, consolidate and simplify the management of account information, improve the security of authentication and directory services, and enforce consistent password and account policies.

Role-based access rights

Centrify enables you to define and manage access rights and role definitions, restrict which users can do what on specific sets of computers or during specific periods of time, and control and restrict access to administrative privileges.

Delegation of authority

Centrify enables you to delegate administrative activity on a task-by-task basis. By delegating individual tasks to specific users or groups, you can establish a separation of duties at the level of granularity you require.

Auditing of activity

Centrify enables you to collect and store an audit trail of user activity when and where you want it. With the auditing service, you can selectively capture and analyze only audit trail events or all user and computer activity.

These features can be used together or independently, depending on the type of licenses you purchase and the specific requirements of your organization. For example, some licenses for Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service might enable identity management, access control, and privilege management. Other licenses might enable auditing of user activity and reporting services.

How zones help you organize information

One of the most important aspects of managing computers with Centrify software is the ability to organize computers, users, groups, and other information about your organization into **Centrify zones**. A Centrify zone is a logical object that you create to organize computers, rights, roles, security policies, and other information into logical groups. These logical groups can be based on any organizing principle you find useful. For example, you can use zones to describe natural administrative boundaries within your organization, such as different lines of business, functional departments, or geographic locations. You can also use zones to isolate computers that share a common attribute, such the same operating system.

Zones provide the first level of refinement for access control, privilege management, and the delegation of administrative authority. For example, you can use zones to create logical groups of computers to achieve the following goals:

- Control who can log on to specific computers.
- Grant elevated rights or restrict what users can do on specific computers.
- Manage role definitions, including availability and auditing rules, and role assignments on specific computers.

- Delegate administrative tasks to implement “separation of duties” management policies.

You can also create zones in a hierarchical structure of parent and child zones to enable the inheritance of profile attributes, rights, roles, and role assignments from one zone to another or to restrict local or remote access to specific computers for specific users or groups.

Because zones enable you to grant specific rights to users in specific roles on specific computers, you can use zones as the first level of refinement for controlling who has access to which computers, where administrative privileges are granted, and when administrative privileges can be used.

You can also use zones to establish an appropriate separation of duties by delegating specific administrative tasks to specific users or groups on a zone-by-zone basis. With zones, administrators can be given the authority to manage a given set of computers and users without granting them permission to perform actions on computers in other zones or giving them access to other Active Directory objects.

Improving security: access and privilege management

Centrify provides its identity management, access control, and privilege management features for Linux and UNIX computers through a combination of features provided by Access Manager and by the Centrify agent on the computers you want to manage.

You can install Access Manager and related management tools on one or more Windows computers. For example, the central console for performing most identity management, access control, and privilege management tasks is Access Manager. From Access Manager, you can perform all of the following common administrative tasks:

- Define and manage identity attributes for the Active Directory users who need access to Linux and UNIX computers.
- Import and migrate UNIX users, groups, and network information from local configuration files and NIS maps.
- Define and manage rights that allow users to run command-line programs, PAM applications, and secure shell operations.

- Select rights to create role-based access control role definitions and assign those roles to the appropriate users and groups.
- Delegate administrative tasks and control the specific permissions granted to users who are managing the computers in your organization.

For example, you can use Access Manager to delegate specific administrative tasks—such as the ability to add and remove users or assign roles—to a particular user or group. As an administrator, you can also use Access Manager to configure roles that have specific start and expiration dates or that limit the availability of a role to specific days of the week or hours of the day. You can use zones in combination with rights and roles to restrict or grant access to specific Linux and UNIX computers in your organization.

Through the use of zones and roles, Centrify provides granular control over **who** can do **what**, and control over **where** and **when** those users should be granted elevated privileges.

Consolidating user account information

Centrify enables you to consolidate all of your user and group account information in a single repository. By consolidating user account information, you can improve IT efficiency and overall operational security. For example, you can automate the provisioning of new accounts and the elimination of accounts that are no longer used without changes to your existing infrastructure or processes.

A single repository also enables you to establish consistent password policies for all of the computers you manage. For example, you can enforce consistent rules for password complexity and minimum length for all users on all computers. A single repository also benefits users, who only have to remember one password, regardless of the computer they use.

By using Centrify zones and override controls, you can migrate your entire user population without modifying any existing account attributes. For example, you can map multiple UNIX profiles with different identity attributes to a single user account, or resolve conflicts if the profiles for different users have the same identity attributes. This flexibility ensures that you can migrate legacy user accounts without changing any existing profile attributes, so that all of the existing directory and file ownership remains unchanged.

Over time, you can then continue to improve organizational security by eliminating legacy identity stores, directories, and databases, including all locally managed `/etc/passwd` files and local user accounts.

Defining role-based access rights

Role-based access rights are more flexible than UNIX group membership rights and easier to define than user specifications in a `sudoers` configuration file. Role-based access rights can be narrowly applied or broadly inherited across any number of computers. You can restrict when role-based rights can be used by defining roles that are available only on certain days of the week or only during specific hours of the day. You can also make role assignments temporary by setting a date and time for the assignment to start or expire. For example, you might given the user Jonah elevated privileges to run administrative commands in the Backup Operators role for a period of two weeks while the primary backup administrator is on vacation.

Role-based access rights also prevent password sharing for privileged accounts, helping to ensure accountability. Users who need to run privileged commands can either temporarily elevate their privileges in an unrestricted login shell or be required to run the commands in a tightly controlled restricted shell without being prompted to provide the administrative password. All of their privileged or restricted shell activity can be traced to the account they used to log on.

Improving accountability: auditing user activity

Centrify provides its auditing and analysis features through a combination of auditing components on Windows computers and the auditing features of the Centrify agent on the computers you manage. The auditing service includes several components to support the multi-tier architecture of the auditing infrastructure. These components are installed on Windows computers to enable you to collect and store detailed information about user activity.

The central console for configuring the auditing infrastructure and managing audit-related features is Audit Manager. From Audit Manager, you can perform the following common administrative tasks:

- View the status of all audited computers and the other components of the auditing infrastructure.

- Manage the scope and security for auditing-related activity.
- Set permissions for the tasks granted to specific auditors.

There is also a separate Audit Analyzer console for searching and replaying captured activity.

Why auditing user activity is important

Just as it is important to protect assets and resources from unauthorized access, it is equally important to track what the users who have permission to access those resources have done. For the users who have privileged access to computers and applications with sensitive information, auditing helps ensure accountability and improve regulatory compliance. With the audit and monitoring service, you can capture detailed information about user activity and all of the events that occurred while a user was logged on to an audited computer.

If you choose to enable auditing on Linux or UNIX computers, the Centrify agent on that computer starts recording user activity as soon as a user logs on. The agent continues recording until the user logs out or the computer is locked because of inactivity. The user activity captured includes an audit trail of the actions a user has taken and a keystroke record of the text that was entered (`stdin`) and the results that were displayed (`stdout` and `stderr`). The information recorded while a user is logged on—which is called a **session**—is collected as it happens, so you can monitor computers for suspicious activity or troubleshoot problems immediately after they occur.

Reviewing user activity

When you audit user activity on a computer, the information is transferred to a Microsoft SQL Server database so that it is available for review and follow-up. Because sessions and audit trail events are stored in the database, you can create queries and reports to find information of interest. For example, you can search the stored user sessions to look for policy violations, command-line execution errors, or malicious activity that may have led to a service degradation or an outage.

In addition to saving the input and output recorded, sessions provide a summary of actions taken so that you can scan for potentially interesting or damaging actions without playing back a complete session. After you select a session of interest in Audit Analyzer, the console displays a list of commands in the order in which the user executed them. You can then select any command in the list to start viewing the session beginning with that action. For example, if the user ran a command that reports credit card information, you can scan the list of commands for the command that accesses credit card information and begin reviewing what happened in the session from that time on.

Using access and auditing features together

You can use access-related features and components without auditing if you aren’t interested in collecting and storing information about session activities. You can also deploy auditing-related features and components without access control and privilege management features if you are only interested in auditing user activity on Linux and UNIX computers. However, you can recognize the most value from Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service by using all of the services as an integrated solution for managing elevated privileges and ensuring accountability and regulatory compliance across all platforms in your organization.

Enabling access control without auditing on a managed computer

If you only enable access control features, the agent enforces the role-based privileges that enable users to log on, access PAM-based application, and run administrative or restricted shell commands. All of the role-based activity is traceable to the user’s own account credentials. However, the audit trail of user activity is only recorded in the computer’s local system log (`syslog`) facility. Information that is only stored in a computer’s `syslog` facility can be more difficult to monitor and query than information stored in a central repository such as Microsoft SQL Server database.

Enabling auditing without access control on a managed computer

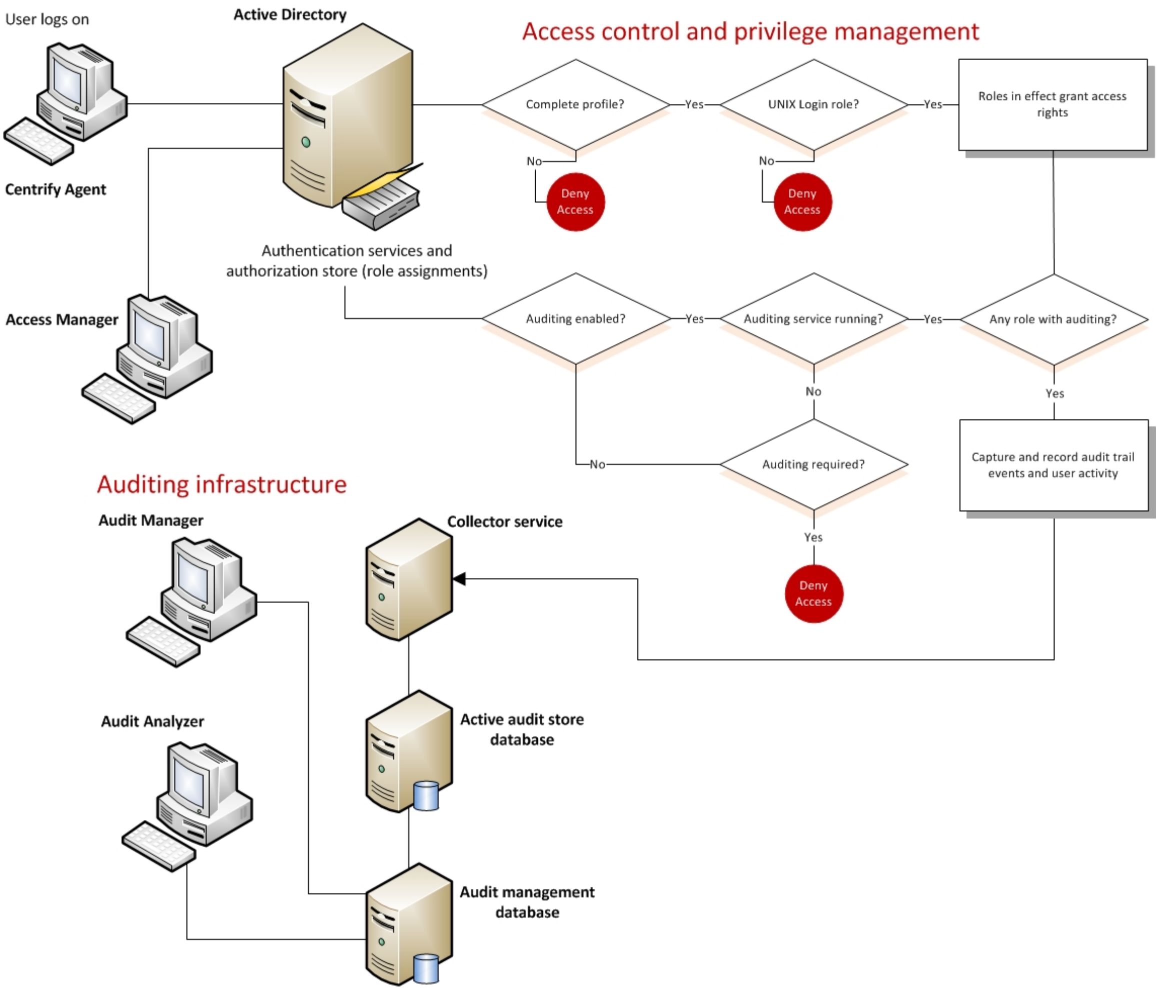
If you only enable auditing, the agent captures detailed information about the command input and output in the login shell of the managed computer. All of the activity is stored in the Microsoft SQL Server database and available to you for queries and reports. However, there’s no role-based enforcement of what activity is allowed on the audited computer.

Enabling access control and auditing on a managed computer

If you use the infrastructure access management and auditing services together, you can define role-based access rights, restrict when and where roles are available, identify roles that should be audited, trace activity when roles with elevated permissions are selected and used, and play back session activity based on the criteria you choose.

By combining access management and auditing on the same computer, you can have an audit trail and, optionally, a video record of all actions performed with elevated privileges. For example, when you deploy access management, users must be assigned to a role with permission to log on. If they are allowed to log on and auditing is deployed, the agent begins auditing their activity. If a user accesses a PAM-based application or executes a privileged command, the action is recorded and can be traced back to the account used to log on.

The following illustration provides a simplified view of the architecture and flow of data when you deploy components for access control, privilege management, and auditing on a Linux or UNIX computer.



However, auditing requires database storage for the audited sessions audit trail events. Auditing also requires additional management of the network connections used to collect and transfer audit-related information from computers being audited to one or more databases where the sessions and audit trail events are stored. If you plan to use the infrastructure access management and auditing services together, you also need to decide which roles should require auditing and which features to enable on each computer you want to manage. In most cases, you choose whether to enable access control features, auditing features, or both feature sets when you install the agent on a computer.

Although it is not depicted in the illustration, you do not have to enable the auditing service to record audit trail events locally for successful or failed operations. By using the auditing service, however, you can store the audited sessions and audit trail events in a database and report on specific types of activity, such as the execution of privileged commands or access to applications and information that must be kept secure. With auditing enabled, the audit trail and the user activity are available for display, querying, and analysis from any computer where you install Audit Analyzer. Through rights and roles you can restrict access to sensitive information and control who can run commands with elevated privileges or perform administrative tasks. Through queries and reports, you can track all of the activity taking place—by user, computer, the time the activity took place, the role that was used, the command that was executed, or other criteria—to verify that only authorized users are performing authorized tasks and to investigate and correct any unauthorized access anywhere in your organization.

For complete information about setting up and managing an audit installation, see the *Auditing Administrator's Guide*.

Managing zones and delegating administrative tasks

Zones are the key component for organizing account profiles, identity attributes, role-based access rights, and role assignments. Zones also enable you to establish logical administrative boundaries and delegate specific administrative tasks to the appropriate users and groups for Linux and UNIX computers. This chapter describes the different types of zones and how to use Access Manager to create and manage zones, modify zone properties, and delegate administrative tasks to other users and groups in your organization.

Starting Access Manager for the first time

The first time you start Access Manager, you can use the Setup Wizard to prepare the Active Directory forest with organizational units and containers for Centrify objects. From the Setup Wizard, you can create either the recommended deployment structure or a custom deployment structure and set all of the appropriate permissions for the objects automatically. If you skip this initial configuration, you can rerun the Setup Wizard at a later time or create organizational units and containers manually. At a minimum, however, you need to select a location in Active Directory for license keys and zones. For more information about the recommended organizational units and permissions, see the *Planning and Deployment Guide*.

What to do before updating Active Directory

Before you use Access Manager the first time, you should contact the Active Directory administrator to determine the appropriate location for the deployment structure and whether you have the appropriate rights for completing this task. The specific administrative rights required for this task depend on the policies of your organization and who has permission to create `classStore` and parent and child container objects in Active Directory.

Rights required for this task

If you don’t have administrative rights to create container objects in Active Directory, a domain administrator in the forest root domain can run the Setup Wizard or manually create the container objects and set the rights on those objects to allow other users to complete the initial configuration without being members of an administrative group.

The following table describes the minimum rights that must be granted on manually created container objects for other users to successfully complete the configuration with the Setup Wizard.

This target object	Requires these permissions	Applied to
`Licenses` container	■ Read all properties ■ Create classStore objects ■ Modify permissions	This object only
	■ Write Description property ■ Write displayName property	This object and all child objects
	By default, all Authenticated Users have read and list contents permission for the Licenses container and all of its child objects.	
`Zones` container	■ Read all properties ■ Create classStore objects ■ Create Container objects	This object only
	■ Write displayName property	This object and all child objects

If you are a domain administrator and manually creating the container objects, you should add a security group for Zone Administrators to Active Directory. Set the following permissions on the parent Zones container to allow other users to manage zones.

This target object	Requires these permissions	Applied to
`Zones` container	■ Read all properties ■ Create Container objects ■ Delete Container objects	This object only
	■ Write displayName property	This object and all child objects

Who should perform this task

A Windows Active Directory administrator performs this task, depending on your organization's policies, by running the Setup Wizard or by manually creating container objects and notifying another user of the location of the container objects. The user who runs the Setup Wizard must be granted the rights required to create `classStore` objects.

How often you should perform this task

In most organizations, you only do this once for an Active Directory forest. However, if you want to create more than one administrative boundary, you can create additional parent containers as needed.

Steps for completing this task

The following instructions illustrate how to run the Setup Wizard from Access Manager.

To update Active Directory using Access Manager:

1. Open Access Manager.
2. Verify the name of the domain controller and the user credentials for connecting to the forest, then click **OK**.
3. At the Welcome page, click **Next**.

4. Select **Use currently connected user credentials** to use your current log on account or select **Specify alternate user credentials** and type a user name and password, then click **Next**.
5. Select **Generate the Centrify recommended deployment structure** if you want to create all of the containers for the recommended deployment structure automatically.

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

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

   If you are generating a default or custom deployment structure, verify the successful execution of the script that creates the structure, then click **Next** to continue.
6. Verify the parent container for licenses is in the top-level Centrify container if you are using the default deployment structure or the container of your choice, then click **Next**.

   You can add other Licenses containers in other locations later using the Manage Licenses dialog box.
7. Review the permission requirements for the container, then click **Yes** to continue.
8. Type or copy and paste the license key you received, then click **Add**.

   If you received multiple license keys, add each key to the list of installed licenses, then click **Next**. If you received license keys in a text file, click **Import** to import the keys directly from the file instead of adding the keys individually, then click **Next**.
9. Verify the **Create default zone container** option is selected and the parent container for zones is in the top-level Centrify container or the container of your choice, then click **Next**.

If you run the Setup Wizard at any time after the initial creation of the Zones container, this step displays the **Change default zone container** option and the current container location. Select this option and click Browse to change the default container for zones, then click **Next**.

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

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

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

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

    This option prevents Centrify profile information from being left "orphaned" when changes are made to Active Directory objects such as users and groups. This option is not selected by default because it requires you to be a member of Enterprise Admins or Domain Admins group for the forest root domain.

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

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

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

What to do next

Create at least one parent zone.

• • • • • •

Where you can find additional information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- How zones help you organize information
- Improving security: access and privilege management
- Improving accountability: auditing user activity

Preparing to create zones

As discussed in How zones help you organize information, Centrify zones help you organize computers, users, groups, access rights and other information into logical groups similar to Active Directory organizational units or Network Information Service (NIS) domains. You have several options when choosing the type of zone to create, and the type of zone you select depends entirely on what your organization needs. The first decision to make is the type of zone to create:

- Hierarchical, which is the default and supports inheritance and overrides.
- Classic, which is backward-compatible to support older versions of the Centrify agent.
- SFU, which supports the Microsoft Services for UNIX schema and rarely used.
- Auto Zone, which is a simplified "zone" for computers to join when you don't need any control over profiles, access rights, or roles and role assignments.

With the exception of SFU zones, you can mix and match any combination of zone types in the same Active Directory forest, as needed. For example, you can create one or more classic zones to support legacy agents, an Auto Zone for a group of computers that don't require the management of identity attributes or access rights, and hierarchical zones for the computers for which you want to actively manage access rights and privileges.

Creating hierarchical zones

Hierarchical zones enable you to establish parent-child zone relationships, allowing profile attributes, rights, role definitions, and role assignments to be inherited down the zone hierarchy. In most cases, you define information in a parent zone so that is available in one or more child zones, as needed. At any point in the zone hierarchy, you can choose to use or override information from a parent zone.

You should use hierarchical zones if your organization has any of the following requirements:

- You have existing user and group profiles that must be migrated with legacy identity attributes to maintain existing file ownership.
- You have user and group profiles that have conflicting identity attributes on different computers.
- You have users and groups that require different role-based access rights, privileges, and role assignments on different sets of computers.

If you are using hierarchical zones, you can use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for login access to Centrify-managed Linux and UNIX computers and for privileged command execution in hierarchical zones, classic zones, and Auto Zone. However, some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multi-factor authentication, see "Preparing to use multi-factor authentication" and the *Multi-factor Authentication Quick Start Guide*.

Creating classic zones

Classic zones do not support inheritance or overrides and have other limitations in how they support role-based access rights. For example, in classic zones, authorization is disabled by default, and must be consciously enabled on a zone-by-zone basis before any role-based access rights or privileges can be configured or assigned.

You should only create new classic zones if your organization has any of the following requirements:

- You must support older versions of the Centrify UNIX agent.
- You have a user population with very few or no identity attribute conflicts.
- You have little or no need to centrally manage access rights and privileges.

If you are using classic zones, you cannot use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for access to Centrify-managed Linux and UNIX computers and for privileged command execution in classic zones. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multi-factor authentication, see "Preparing to use multi-factor authentication" and the *Multi-factor Authentication Quick Start Guide*.

Creating an Auto Zone

Most organizations that deploy the Centrify agent on Linux or UNIX computers have an existing user population to migrate to Active Directory, and hierarchical zones make the most sense. However, multiple zones are not required for all situations. You can greatly reduce the time required and complexity of your deployment if a single zone suits your organization's needs. This type of zone is created automatically when computers join the domain using the `--workstation` option.

An Auto Zone automatically enables all of the users and groups in an Active Directory forest to become valid users and groups on the Linux and UNIX computers that join the Auto Zone.Their profiles are generated automatically and there's no need to manage account profiles, access rights, privileges, or delegated administrative tasks.

You should only use the Auto Zone option if your organization meets the following requirements:

- You are not migrating an existing user population.
- You want to automatically generate profiles for all or most Active Directory users and groups without managing identity attributes.
- You don't want to configure and manage role-based access rights and privileges or role assignments.

If you are using an Auto Zone, you cannot use the local account management feature as described in Managing account profiles and identity attributes

You can configure multi-factor authentication for both licensed and Express agents to control access to Centrify-managed Linux and UNIX computers. For licensed agents, you can also require multi-factor authentication to run privileged commands in an Auto Zone. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For details about configuring multi-factor authentication, see "Preparing to use multi-factor authentication" and the *Multi-factor Authentication Quick Start Guide*.

Creating a new parent zone

In most cases, you design a basic zone structure as part of the deployment process. After the initial deployment, you can create new hierarchical zones any time you have new administrative boundaries. For example, if you acquire another organization, add offices that are managed by a different group, or restructure the organization along different functional lines, you are likely to need new zones.

You can create as many parent zones as you need. You must create at least one new zone before you begin adding Linux and UNIX computers to the Active Directory domain, unless you are joining with the `--workstation` option.

What to do before creating a new parent zone

Before you can create parent zones, you must have installed Access Manager and run the Setup Wizard. You should also have a basic zone design that describes how you are organizing information, for example, whether you are using one top-level parent zone or more than one parent zone. You should also decide whether to create the new zone in the default Zones container object or in another container or organizational units within Active Directory. There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new zones, your user account must be a domain user with the following permissions:

Select this target object	To apply these permissions
Parent container for new zones, for example: `domain /Centrify/Zones`	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: ▪ Create Container Objects ▪ Create Organizational Unit Objects **Note** Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: ▪ Create group objects ▪ Write Description property

Note: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who should perform this task

A Windows domain administrator performs this task, depending on your organization's policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are fully deployed, you create new zones infrequently to address changes to your organization.

Steps for completing this task

The following instructions illustrate how to create a new parent zone using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a new parent zone using Access Manager:

1. Open Access Manager.
2. Select Zones, right-click, then click **Create New Zone**.
3. Type the zone name and, optionally, a longer description of the zone.

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

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

   - Create a zone in a new location to separate administrative activity for different groups of administrators.
   - Create a zone as an organizational unit because you want to assign a Group Policy Object to the zone.
   - Create a classic or SFU zone to support legacy Centrify agents or to store data using the Microsoft Services for UNIX schema.

   For additional information about any field in the new zone wizard, you can press F1 to view the context-sensitive help.
4. Review information about the zone you are creating, then click **Finish**.

What to do next

After you create a new parent zone, you might want to create its child zones.

Where you can find additional information

If you want to learn more about the importance and benefits of using zones, see the following topics for additional information:

- How zones help you organize information
- Preparing to create zones

Creating child zones

The primary reason for creating child zones is to inherit profile attributes, role definitions, and role assignments from a parent zone. You can then use the child zone to override the specific profile attributes that might be different on a given set of Linux and UNIX computers than you have defined in the parent zone. Less often, you might want to use a child zone to override specific access rights, role definitions, or roles assignments that you have made in a parent zone. For example, if you have created a role definitions that allows a user to run a specific application with administrative privileges in a parent zone, you can use child zones to limit the scope of that right to specific subsets of computers.

What to do before creating child zones

Before you create child zones, you must have installed Access Manager, run the Setup Wizard to create the Zones container, and created at least one parent zone. You should also have a basic zone design that describes the zone hierarchy for the child zone. There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups through the Zone Delegation Wizard. To create new child zones, your user account must be a domain user with the following permissions:

Select this target object	To apply these permissions
Container for the parent zone, for example if the parent zone is `berlin`: `domain /MyOU/Zones/berlin`	On the **Object** tab, select **Allow** to apply the following permission to this object and all child objects: ▪ Create Container Objects ▪ Create Organizational Unit Objects **Note** Both permissions are required if you want to allow zones to be created as either container objects or organizational unit objects.
Parent container for Computers in the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: ▪ Create group objects ▪ Write Description property These permissions are only needed if you are supporting “agentless” authentication in the new zone.

Note: If the Active Directory administrator manually sets the permissions required to create zones, you should verify that the account also has permission to add an authorization store, define rights and roles, and manage role assignments.

Who should perform this task

A Windows administrator performs this task, depending on your organization’s policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are fully deployed, you create new child zones infrequently to address changes to the scope of ownership and administrative tasks.

Steps for completing this task

The following instructions illustrate how to create a new child zone using Access Manager. Examples of scripts that use the Access Module for Windows

PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a new child zone using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new child zone.
3. Right-click, then click **Create Child Zone**.
4. Type the zone name and, optionally, a longer description of the zone.

   Because this is a child zone, you should use the default parent container and container type, then click **Next**.
5. Review information about the child zone, then click **Finish**.

Opening and closing zones

Because properties and objects are organized into zones, you must open a zone to work with its contents. If you open a parent zone, its child zones are also available for you to use by default. If you open a child zone, you can choose whether to open its parent zone.

To open an individual parent or child zone:

1. Open Access Manager.
2. Select Zones, right-click, then click **Open Zone**.
3. Type all or part of the name of the zone you want to open, then click **Find Now**.
4. Select the zone to open from the list of results, then click **OK**. You can use the **CTRL** and **SHIFT** keys to select multiple zones.

Loading all zones

As an alternative to opening individual or parent and child zones manually, you can automatically load all zones in a forest or all zones in a specific container at startup time. If you choose to load all zones, you cannot manually close zones.

To load all zones automatically:

1. Open Access Manager.
2. Select Access Manager, right-click, then click **Options**.
3. On the **Filter Settings** tab, select **Load all zones**, then select **connected forest** to automatically load all zones in the forest or click **Browse** to navigate to specific container.

You should not select the Load all zones option if you want to manually open and close zones for performance reasons.

Closing individual zones

After you open a zone, it stays open during your current sessions unless you close it. If you have a large number of zones, however, you should close any zones you aren't actively working with for better performance.

To close an open zone:

1. Open Access Manager.
2. Expand Zones and select an open parent zone, right-click, then click **Close**.
3. Click **Yes** to confirm that you want to close the zone.

Delegating administrative tasks

If you have created at least one zone, you can give other users and groups permission to perform specific types of administrative tasks within that zone. For example, assume you have created a new zone called `Finance` and you want to give the users who access computers in this zone the permissions required to perform certain kinds of tasks based on their role. You can accomplish this goal by selecting a group or users, then assigning that group or user one or more tasks. For example, in the `Finance` zone, you might want to delegate administrative tasks like this:

- The members of the Active Directory group `FinanceITStaff` are allowed to perform all administrative tasks in the `Finance` zone.
- The members of the Active Directory group `FinanceManagers` are allowed to add, modify, and remove user and group profiles in the `Finance` zone.
- The members of the Active Directory group `FinanceUsers` are allowed to join computers to the `Finance` zone, but perform no other tasks.
- The Active Directory users `jason.ellison` and `noah.stone` are granted permission to manage role assignments in the `Finance` zone.

In most cases, each zone should have at least one Active Directory group that can be delegated to perform all administrative tasks, so that members of that group can manage their own zone. You are not required to create or use a zone administrator group for every zone. However, assigning the management of each zone to a specific user or group creates a natural separation of duties for administrative tasks.

If you delegate control for individual tasks—for example, by assigning only the join computers task to one group and only the add and remove users tasks to another—you should ensure the members of each group know the tasks they are assigned.

You can delegate administrative tasks for parent zones, for child zones, and for individual computers. Because computer-level overrides are essentially single computer zones, you can assign administrative tasks to users and groups at the computer level.

What to do before delegating administrative tasks

Before you delegate administrative tasks for a zone, you must have created at least one zone. For each zone you create, you should also identify at least one user or group that can be delegated to perform all administrative tasks. For example, if you have a `Finance` zone, you might want to create a `Finance Admins` group in Active Directory, then delegate **All** tasks to that group so that members of that group can manage their own zone.

There are no other prerequisites for performing this task.

Rights required for this task

Only the user who creates a zone has full control over the zone and can delegate administrative tasks to other users and groups.

For information about the permissions set when you select different administrative tasks in the Zone Delegation Wizard, see the *Planning and Deployment Guide*.

Who should perform this task

The domain administrator who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. Only the account used to create a zone has full control over the zone's properties and permission to delegate administrative tasks to other users. The user who creates a zone is also the only user who can add NIS maps to the zone. The right to create NIS maps is exclusive to the creator of a zone because it requires permission to create containers in Active Directory. The zone creator can, however, grant other users permission to add, remove, or modify NIS map entries.

How often you should perform this task

In most organizations, you delegate administrative tasks any time you create a new zone. You also might change the delegation to change the either tasks assigned or the users and groups that have been assigned specific tasks periodically to address changes to your organization. For example, if an existing zone administrator takes over new responsibilities or leaves the organization, you might need to delegate additional tasks or select a different user or group to perform administrative tasks.

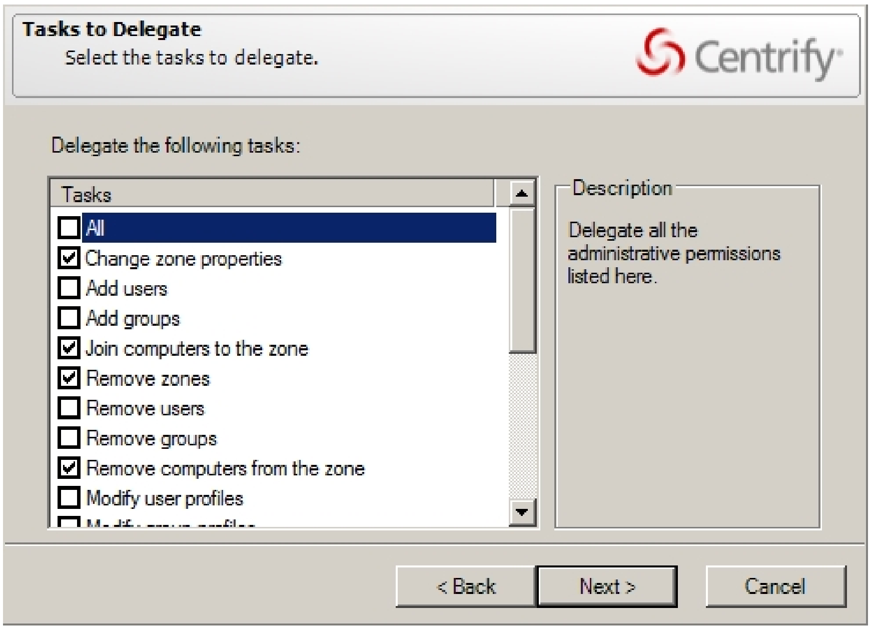
Steps for completing this task

The following instructions illustrate how to delegate zone administration tasks to a user, group, or computer using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To delegate administrative tasks to specific users and groups in a zone:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name for which you want to delegate administrative tasks.
3. Right-click, then click **Delegate Zone Control**.
4. Click **Add** to find the users, groups, or computer accounts to which you want to delegate specific tasks.
5. Select the type of account—**User**, **Group**, or **Computer**—to search for, type all or part of the account name, then click **Find Now**.
6. Select one or more accounts from the list of results, then click **OK**.
7. When you are finished adding users and groups to which you want to assign administrative tasks, click **Next**.
8. Select the tasks you want to delegate to the user or group, then click **Next**.

   For example, if you want all of the members of the group you selected in the previous step to be able perform all administrative tasks for a zone, check the **All** task. To restrict the administrative tasks a user or group can perform, select only those specific tasks.

   Tasks to Delegate
   Select the tasks to delegate.

   Delegate the following tasks:

   Tasks
   - ☐ All
   - ☑ Change zone properties
   - ☐ Add users
   - ☐ Add groups
   - ☑ Join computers to the zone
   - ☑ Remove zones
   - ☐ Remove users
   - ☐ Remove groups
   - ☑ Remove computers from the zone
   - ☐ Modify user profiles
   - ☐ Modify group profiles

   Description
   Delegate all the administrative permissions listed here.

9. Review your selections, then click **Finish**.

If you have delegate administrative tasks to one or more groups that have members logged on, you should notify the group members to log out and log back on before they attempt to perform the administrative tasks assigned to the group.

Changing zone properties

After you create a zone, you can change its zone properties at any time. For example, if you want to change the parent zone for a child zone, you can do so by modifying the child zone's properties. Depending on whether you are viewing a classic, hierarchical, or SFU zone and the components you have installed, you might see and be able to set different zone properties.

To display the properties for a zone:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.

   If the zone you have selected is a hierarchical zone, the properties are organized on the following tabs.

Use this tab	To do this
General	View and set general information about the selected zone, including the location of the zone in Active Directory, the zone type, and the zone description. For additional details about general properties, see the following topics: - Changing the zone description - Changing the parent zone or location of a zone - Setting the master domain controller for a zone - Selecting a license container for a zone - Adding support for agentless clients - Setting custom permissions for a zone
Platform	View and set the identity platform instance to use for the selected zone. For additional details about setting identity platform properties, see the following topic: - Selecting a identity platform instance for a zone
User Defaults	Set default values for user profile attributes in the selected zone. For additional details about user default properties, see the following topic: - Setting user defaults
Group Defaults	Set default values for group profile attributes in the selected zone. For additional details about group default properties, see the following topic: - Setting group defaults
Variables	Add or edit user-defined variables or override the default values of predefined variables in the selected zone. For additional details about zone variables, see the following topic: - Configuring variables for a zone
Provisioning	Configure automated provisioning for user and group profiles if you have the Zone Provisioning Agent installed on the local computer. For additional details about provisioning properties, see the following topic: - Configuring automated provisioning The Provisioning tab is only displayed if the Zone Provisioning Agent is installed. For detailed information about configuring automated provisioning, see the *Planning and Deployment Guide*.

Changing the zone description

You can set or change the optional description for a zone at any time. For example, if you didn't specify a description when you created the zone or if there have been changes in your organization that warrant a change in the description of a zone, you can modify the Description field to make the change.

To change the zone description

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Type a description for the zone in the Description field, then click **OK**.

Changing the parent zone or location of a zone

From Access Manager, you can make any existing hierarchical zone the child of another zone or make any child zone a new parent zone by dragging and dropping the zone into a new location or by changing the Parent zone field on the zone's General properties tab.

Selecting the default location when moving a zone

If you make changes to the zone hierarchy, Access Manager prompts you to specify the new Active Directory location for the zone. In most cases, you should accept the default location for the zone you are moving. The default Active Directory location will be either:

- The **new parent zone** container if you are moving a child zone from one parent to another or if you moving a parent zone to become a child zone.
- The default **Zones** container you created the first time you started Access Manager if you are making a child zone a new top-level parent zone.

You are not required to accept the default Active Directory location when changing the zone hierarchy. If you select a different Active Directory location for the zone, however, you should note the location and whether the zone you are moving is now a parent or a child zone. If the zone structure displayed in Access Manager is different from the zone container structure you are using in

Active Directory, you might find unexpected problems with inheritance and overrides, with modifying zone properties, or with deleting zones.

Moving a zone without changing its Active Directory location

When you are prompted to specify the Active Directory location for a zone you are moving, you have the option to select **No** and leave the current Active Directory location unchanged. If you change the parent zone without changing the Active Directory location for a zone, you should note that the location does not reflect the zone hierarchy. In rare cases, you might find it useful to leave the Active Directory location unchanged but doing so might make it more difficult to locate the zone object at a later time.

Restarting the agent after moving a zone

If you change the location for a zone in Active Directory, you must restart the Centrify UNIX agent on the computers in that zone so that they recognize the new zone location.

After you move the ZoneName object to a new parent container or organizational unit, run the following command to restart the Centrify UNIX agent on the computers in the zone:

```
/usr/share/centrifydc/bin/centrifydc restart
```

To move a zone to a new parent by changing properties

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. For the Parent zone field, click **Browse** to find and select the zone to use as the parent, then click **OK**.
5. Click **OK** to save the new zone properties.
6. In the Move Zone dialog, verify the location selected for the **Yes, move to** option to accept the default location, then click **OK**.

   In rare cases, you might want to click **Browse** to select a different Active Directory location for the zone you are moving, or select **No**, then click **OK** to keep the zone in its original location.

Setting the master domain controller for a zone

In most cases, computers connect to the first available Active Directory domain controller and it is not necessary to specify the master domain controller to use for a zone. In some cases, however, you might want to identify a specific domain controller to use for a zone to prevent connections from other domain controllers from adding or removing users and groups in that zone.

To prevent connections from other domain controllers, you can set the Master domain controller field to the fully-qualified name of the domain controller you want to use. After you identify a master domain controller, administrators who connect to the zone using any other domain controller will not be able to make changes to the zone.

If you have multiple administrators managing any zones, you should notify them before setting or changing the master domain controller. You should also make this change while all other administrators are logged off. Depending how long it takes for replication to complete for all of the domain controllers in the Active Directory forest, you might want to schedule this change for a time when no administrators need access to zone information.

To change the master domain controller

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to change the master domain controller.

   You can use `Shift-Click` or `Ctrl-Click` to select multiple zone names.
3. Right-click, then click **Change Master Domain Controller**.
4. Type the fully-qualified domain name for the new domain controller, then click **OK**.
5. Click **Yes** to confirm that you want to change the master domain controller for the zone.

You should avoid changing from one master domain controller to another, if possible. Changing the master domain controller requires you to wait for replication to complete to see up-to-date zone information or modify information in the selected zone. In some cases, however, changing the master domain controller might be unavoidable. For example, if there are zones connecting to a master domain controller that has a hardware failure or must be taken offline for maintenance, you will need to configure a new master domain controller for the zones to use.

If you change the master domain controller, you should run the Analyze command afterwards to check the Active Directory forest and verify that no duplicate UIDs or GIDs have been introduced.

Selecting a license container for a zone

By default, zones are configured to use any available license container in the forest. In most cases, the container used is the default **Licenses** container you created the first time you started Access Manager. If you have more than one Licenses container, you might want to select a specific license container for a set of computers in the one zone and a different license container for a set of computers in another zone. For example, you might want to select separate Licenses containers for the zones associated with two different business units.

To use a specific license container for a zone, you can type the path to a new container object in the License container field.

To use a specific license container for a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Select a specific license container from the list of available License container, then click **OK**.

For more information about licenses keys and using multiple license containers, see the *License Management Administrator's Guide*.

Adding support for agentless clients

If you are using the Centrify Network Information Service (`adnisd`) on a managed computer to respond to NIS client requests from computers where the Centrify agent cannot be installed, you can configure one or more zones to act as the NIS domain for those client requests.

To add support for agentless NIS clients in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties** to display the General tab.
4. Select the **Support agentless client** option.
5. Select the Active Directory attribute you want to use to store the password hash and verify the zone name is the NIS domain name you want to use or type a new name, then click **OK**.

For more information about installing and using the Centrify Network Information Service (`adnisd`) to respond to NIS client requests and configuring agentless clients, see the *Network Information Service Administrator's Guide*.

Setting custom permissions for a zone

For convenience, you can access Permissions for a zone directly from the zone properties General tab. You can then allow or deny basic permissions—such as Read and Write permissions—to specific users and groups or click **Advanced** to set more granular permissions on a zone.

Selecting a identity platform instance for a zone

In most cases, the identity platform instance property is set automatically when you register a connector for Privileged Access Service. If you have access to more than one identity platform instance—for example, if you have more than one customer identifier, you can select the URL for a specific instance from the zone properties.

To select a identity platform instance for a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.

4. Click the **Platform** tab.
5. Verify the identity platform instance URL is the customer-specific URL you want to use, or click **Browse** to select the URL for a different customer-specific identity platform instance.

   Child zones inherit the identity platform instance property from their parent zone. If you are viewing properties for a child zone, you can select **Override trusted identity platform instance** then click **Browse** to select a different identity platform instance for the child zone.

   For details about installing and configuring a connector, see "Preparing to use multi-factor authentication."
6. Click **OK** to confirm the identity platform instance selected.

Configuring default values for a zone

You can configure default settings for user and group profiles that are added to the zone. The user and group defaults you configure can include predefined variables that populate the user or group profile by using Active Directory attributes or settings configured on individual managed computers.

By specifying user default and group default settings, you can simplify the process of adding user and group profiles to child zones. For example, you can define a default user profile that uses the `sAMAccountName` attribute for a user's UNIX login name. All users who are added to the zone are then automatically assigned a UNIX login name based on their `sAMAccountName`. If you define the default attributes in a parent zone, they can also be inherited in all of the child zones under that parent and only overridden where other values are explicitly required.

Setting user defaults

When you create a zone, it includes a default set of user profile attributes. In most cases, there's no need to modify any of the default settings unless you want to define partial profiles in a parent zone that will be manually completed in child zones. For example, the default setting for the numeric user identifier (UID) is an automatically generated UID based on the user's globally unique security identifier (`SID`). This setting ensures all users who are added to the zone are assigned a unique UID for the entire forest.

If you define a default value for any user profile attribute, that value is used to populate the user profile displayed when you add users to the selected zone.

When you add a user to the zone, you can accept the default profile attributes or override any of the default attributes displayed.

To view or modify the default user profile in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **User Defaults** tab.
5. Review the default settings and modify any of the defaults, if needed.

   For most organizations, the default settings are appropriate. For example, the Active Directory `sAMAccountName` attribute most closely resembles the most common format for the UNIX login name and an automatically generated UID ensures that all new users have a unique UID in the forest. For more information about the attribute fields or the default values, press F1 to view the context-sensitive help.
6. Click **OK**.

For more information about using default values, see Creating user profiles for Active Directory users. For more information about using predefined or custom variables in user profiles, see Setting runtime variables in user profiles.

Setting group defaults

When you create a zone, it includes a default set of group profile attributes. In most cases, there’s no need to modify the default settings for groups unless you are manually assigning numeric group identifiers (GID) or using the Apple algorithm for generating the GID.

If you define a default value for a group attribute, that value is used to populate the group profile displayed when you add groups to the selected zone. When you add a group to the zone, you can accept the default profile attributes or override any of the default attributes displayed.

To view or modify the default group profile in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Group Defaults** tab.
5. Review the default settings and modify any of the defaults, if needed.

   For most organizations, the default settings are appropriate. For example, the Active Directory `sAMAccountName` attribute most closely resembles the most common format for the group name and an automatically generated GID ensures that all new group have a unique GID in the forest. For more information about the attribute fields or the default values, press F1 to view the context-sensitive help.
6. Click **OK**.

For more information about using default values, see Creating group profiles for Active Directory groups. For more information about using predefined or custom variables in user profiles, see Setting runtime variables in user profiles.

Configuring variables for a zone

Predefined and custom variables enable you to generate user profiles and group profiles using Active Directory properties or properties defined on managed computers.

You can add custom runtime variables, or override the definition for predefined variables, in a zone by modifying the zone properties. Runtime variables are resolved by the agent when a computer joins a zone. The default user profile settings use predefined runtime variables in place of specific values for the GECOS, Home directory, and Shell attributes.

Zone variables and their definitions are inherited down the zone hierarchy, and can be overridden in a child zone or on individual computers. You can also use configuration parameters to control the value for any variables locally on particular computers. If a value is set in the configuration file, it overrides any values that you set for the zone.

Adding custom runtime variable

In most cases, you don't need to add custom variables to a zone. However, if you have modified the Active Directory schema or want to use custom attributes in user or group profiles, you can add custom variables to the zone to accommodate your changes.

To add a custom variable to a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.
4. Click the **Variables** tab.
5. Click **Add**.
6. Type a variable name and a value, then click **OK**.

   For example, you might want to define a custom variable named `gecos` and set its value to a static string, such as `Engineering-Nova Scotia-Q22`, for a zone.

   Similarly, you might want to add custom variables for different operating systems you support, such as `mac-home` or `aix-shell` for a zone that includes computers with different operating systems. For example, if a zone includes Linux, AIX, and Mac OS X computers, you might have a default profile that uses the predefined variables, but a subset of accounts that use the `mac-home` or `aix-shell` custom variables.
7. Click **OK** to save the properties.

Modifying predefined variable values

In most cases, you don't need to override predefined variable values for a zone. However, if you have created different zones for different operating systems, you might find it useful to modify predefined variable values for those zones to address different operating system requirements.

To modify a predefined variable value in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones, as required, to locate and select the zone name for which you want to display properties.
3. Right-click, then click **Properties**.

4. Click the **Variables** tab.
5. Click **Add**.
6. Type the name of a predefined variable and a value, then click **OK**.

   For example, you might want to change the predefined variable named `home` and set its value to an appropriate home directory for the zone, such as `/export/home` for a zone where all of the computers are Solaris computers, or `/Users` for a zone with only Mac OS X computers. Similarly, you might want to change the predefined variable `shell` to set its value to `/usr/bin/ksh` for a zone with IBM AIX computers.
7. Click **OK** to save the properties.

Editing or removing variables

After you have added custom variables or modified predefined variable values in a zone, you can later select those variables to edit or remove them.

Configuring automated provisioning

The Centrify Zone Provisioning Agent is a separate service that enables automated provisioning and de-provisioning of user and group accounts on a zone-by-zone basis. You can configure the Zone Provisioning Agent to monitor specific Active Directory groups for a zone. If you add or remove Active Directory users or groups in the monitored groups, the Zone Provisioning Agent automatically adds or removes the corresponding user or group profiles in the zone. If you have the Centrify Zone Provisioning Agent installed, you can use the zone properties Provisioning tab to do the following:

- Enable provisioning for users, groups, or both.
- Specify the Active Directory group to base provisioning on.
- Select the method for automatically generating profile attributes for users, groups, or both.

For more detailed information about automated provisioning and using the Zone Provisioning Agent, see the *Planning and Deployment Guide*. For more information about the attribute fields or the options for generating profile attributes, press F1 to view the context-sensitive help.

Renaming a zone

You can rename a zone at any time. For example, if your organization changes how business units are aligned, moves to a new location, or merges with another organization, you might want to update zone names and descriptions to reflect these changes. You might also want to rename zones if your initial deployment did not use a naming convention for new zones, and you want to implement one after you have agents deployed.

What to do before renaming a zone

Before you rename zones, you might want to define and document a naming convention to use for future zones or the reasons for changing the zone name. You should also identify the computers in the zone to be renamed. You must restart the agent on those computers for the new zone name to be recognized. There are no other prerequisites for performing this task.

Rights required for this task

To rename a zone, your user account must be set with the following permissions:

Select this target object	To apply these permissions
Parent container for an individual zone For example, a ZoneName container object, such as: `domain /Zones/arcade`	▪ Write Description ▪ Write name ▪ Write Name These are the minimum permissions required to rename a zone and not allow a user or group to modify any other zone properties. You can set permissions manually, or automatically grant these and other permissions to specific users or groups by selecting the **Change zone properties** task in the Zone Delegation Wizard.

Who should perform this task

A Windows administrator performs this task, depending on your organization’s policies. The user who creates the zone is responsible for delegating administrative tasks to other users or groups, if necessary. In most organizations, this task is done using an account with domain administrator privileges.

How often you should perform this task

After you are deployed, you rename zones only when you need to address organizational changes or to implement or improve the naming conventions you use.

Steps for completing this task

The following instructions illustrate how to rename a zone using Access Manager.

To rename a zone using Access Manager:

1. Open Access Manager.
2. Expand **Zones** to display the list of zones, then expand any child zones in the zone hierarchy until you see the specific zone you want to modify.
3. Select the zone to change, right-click, then click **Rename**.
4. Type the new name and, if needed, any changes to the zone description.
5. Restart all of the Centrify UNIX agents on the computers in the zone you have renamed.

   You do not have to leave and rejoin after changing a zone name. However, you must restart the agent for the name change to take effect on a managed computer. In a terminal window on each managed computer, run the following command:

   ```
   /usr/share/centrifydc/bin/centrifydc restart
   ```
6. You can verify the updated zone name on a local computer by using the `adinfo` command, which includes the joined zone name in its output.

Adding computers to a zone

You can only join a domain by creating a computer account that is either a “zone computer” profile or a “workstation” account that uses Auto Zone. Depending on the tool and operating system you prefer to use, there are several ways you can add a computer account to a zone. For example, if you prefer to create the “zone computer” account from a Linux or UNIX computer:

- You can run the `adjoin` command interactively or in a script and specify the zone as a command line option while joining the Active Directory domain.
- You can use `ADEdit` commands interactively or in a script to add a computer account to a zone before joining the domain.

If you prefer to create the “zone computer” account from a Windows computer:

- You can prepare a computer account in Access Manager before joining the domain.
- You can use the Centrify Access Module for PowerShell cmdlets interactively or in a script to add a computer account to a zone.

Precreating computer accounts using ADEdit or the Centrify Access Module for PowerShell cmdlets is particularly useful if you want to join multiple computers with minimal command line options and if you want to allow the computer account to be used to perform a “self-service” join. For more information about preparing a computer account before joining a domain, see Preparing computer accounts before joining.

For more information about specifying the zone, joining the domain, and modifying computer properties, see Working with managed computers. For information about using Auto Zone, see Using workstation mode and Auto Zone.

Managing licenses

The first time you start Access Manager, you are prompted to create a Licenses container and add or import license keys. You can also add and remove license containers and keys after the initial configuration.

To modify license information

1. Open Access Manager
2. Select Access Manager, right-click, then select **Manage Licenses**.
3. Click Add to add a new license container or license key.
4. Select an existing license container or license key, then click **Remove** to remove that container or key.

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

Reporting zone information

You can access legacy reports from within Access Manager by selecting Access Manager, right-clicking, then selecting **Report Center**. You can also use command-line programs, PowerShell scripts, or ADEdit scripts to report zone information. In most cases, however, you should install and configure Report Services to generate and access reports about the Active Directory domain and your zones.

For details about installing and configuring Report Services, and how to customize and access the reports that generated, see the *Report Administrator's Guide*.

Migrating from classic to hierarchical zones

Classic zones are primarily intended for backward compatibility with older versions of the Centrify agent. If you upgrade the agent to version 5.x or later, you can migrate any or all of your classic zone information into one or more hierarchical zones.

Migrating a classic zone to a hierarchical zone is a multi-step process that requires some initial planning. For example, the first step in the migration changes the zone type but does not change any existing zone information, including the computer accounts that are joined to the zone. To take full advantage of the hierarchical zone after migration, however, it is likely that you will need to modify some of your existing zone information and move computer accounts into different zones.

Preparing for migration

To prepare for the migration of any classic zones, you should first review the existing zone information for “dominant” user and group profiles—that is, profiles with attributes that are common to multiple classic zones. Dominant profiles will help you to identify one or more classic zones that you can use as potential parent zones. A parent zone provides a baseline for the user and group profiles that can be inherited in child zones. The parent zone also enables you to manage rights and role definitions that can be inherited in the child zones you create. If you are able to identify dominant profiles, most of your classic zones will become child zones that inherit information from the parent zone, with specific attribute overrides on a zone-by-zone or computer-by-computer basis, as necessary.

To illustrate how you should analyze your existing environment, assume you have several classic zones to address different profile requirements on different computers, but only two administrative groups that have different policies and procedures for adding users or granting privileges. In this scenario, you might create two parent zones—one for each administrative team—and use child zones or computer overrides to address specific profile attribute differences. If your organization has a single account fulfillment desk that handles all provisioning and access privileges, you might create a single parent zone for managing all or most user and group profiles, then use child zones to manage more granular account privileges.

If you have a “master” classic zone where the most commonly-used profile attributes for most of your users and groups are defined, that zone is a likely candidate to become a hierarchical parent zone. If none of your existing classic zones is suitable to become a parent zone, you should create a new parent zone as described in Creating a new parent zone. The parent zone must exist before you can use the migration utility.

Verifying you have upgraded Access Manager

You can use Access Manager to view and manage any combination of zones. However, the console must be version5.x or later to work with hierarchical zones. You can check the version of the console you have installed by opening Access Manager, clicking Help, then selecting About Access Manager.

• • • • • •

Verifying you have upgraded UNIX agents

The migration utility is a command-line program installed with the Centrify UNIX agent. You must upgrade the agent to version 5.1, or later, on at least one UNIX computer to do any migration. You can verify the agent version by running the `adinfo` command with the `--version` (-v) option.

What the migration utility does

After you have identified at least one classic zone as potential parent zones, you can use the migration utility to convert the classic zone into a hierarchical parent zone. After you make the classic zone a hierarchical zone, you can run the migration utility to make other classic zones into child zones of the parent zone.

During the migration, all of the user and group profiles in the source zone are copied to the specified parent zone. If identical profiles exist in multiple classic zones, the identical profiles become a single profile in the parent zone. If there are user or group profiles in multiple zones with different attribute values—for example, a UID of 10001 in one classic zone, but 10003 in another classic zone, the migration utility creates a single profile for the user in the parent zone, and creates a profile override with the distinct attribute values in each target child zone. Each child zone inherits the base profile from the parent zone but applies the overrides for any attributes that are different in different zones.

The migration utility copies everything else—including rights, roles, role assignments, groups, and NIS maps—into the new child zone for each classic zone being migrated.

Using the migration utility

Centrify provides the command-line program `admigrate` to simplify the process of migrating profiles, rights, roles, role assignments, and NIS maps from a classic zone to a hierarchical zone.

The `admigrate` program is installed by default in the following directory:

```
/usr/share/centrifydc/adedit/admigrate
```

Note that the first zone you migrate becomes the primary source of profile information for the other zones you migrate. You should start with the zone that it contains the most consistent profile attributes.

Note: Admigrate does not migrate classic SFU zones (Ref: CS-28289a) nor zone delegation rights (Ref: IN-90002).

To migrate zone information from a classic to hierarchical zone:

1. Log on to a Linux or UNIX computer running `adclient` and open a terminal window.
2. Open a text editor to create a file with bind information for each domain to which `admigrate` must connect.

   Specify the Active Directory credentials for an account with permission to create child zones, rights, roles, user profiles, and group profiles in the parent zone with one line per domain in the format:

   ```
   bind domain_account_password
   ```

   For example, create a file named `migrate.conf` with information similar to the following:

   ```
   bind finance.acme.com administrator {myP@$swd}
   bind eng.acme.com engadmin {@lt!pas$}
   ```

3. Save and close the file.
4. Run the `admigrate` command.

   ```
   admigrate -in classicZone -z targetZone -hz parentZone -
   config configFile
   ```

For this variable	Specify this information
classicZone	The distinguished name of the classic zone to migrate. For example: `“cn=finance,cn=zones,ou=unix,dc=acme,dc=com”`
targetZone	The distinguished name of the new zone. It can be the same as the existing classic zone name, however the new zone will be a child zone of the specified parent zone, so the distinguished name is different. For example: `“cn=finance,cn=global,cn=zones,ou=unix,dc=acme,dc=com”`

For this variable	Specify this information
parentZone	The parent zone for the migration. The specified zone must be an existing zone. The target zone becomes a child zone of this zone. You can run `admigrate` multiple times and specify the same parent zone and different source and target zones each time to migrate multiple zones to different child zones of this parent. For example: `“cn=global,cn=zones,ou=unix,dc=acme,dc=com”`
configFile	The configuration file to use with the migration. The configuration file is primarily useful to specify bind information if you are migrating zones from domains that are different from the target zone’s domain. The file is a simple text file, for example: `-config admigrate.txt`

For more information about other options you can use when running `admigrate`, see the `man` page for `admigrate`.

The first time you run `admigrate`, the command copies all of the user profiles from the source zone to the parent zone. Everything else defined in the source zone—including groups, rights, role definitions, role assignments, and NIS maps—is copied from the source zone to a new target child zone.

5. Repeat Step 4 for each classic zone you want to migrate as a child of the parent zone.

Sample migration

To illustrate how to use the `admigrate` command, assume you are migrating two classic zones—`finance` and `engineering`—into a new empty parent zone named `global`. For this example, the distinguished name of the classic `finance` zone (the source zone) is this:

```
“cn=finance,cn=zones,ou=unix,dc=test,dc=org”
```

After migration, the distinguished name of the `finance` child zone (the target zone) is this:

```
“cn=finance,cn=global,cn=zones,ou=unix,dc=test,dc=org”
```

To migrate the classic `finance` zone, you would run a command similar to the following:

```
/usr/share/centrifydc/adedit/admigrate \
-in “cn=finance,cn=zones,ou=unix,dc=test,dc=org” \
-z “cn=finance,cn=global,cn=zones,ou=unix,dc=test,dc=org” \
-hz “cn=global,cn=zones,ou=unix,dc=test,dc=org” \
-config ~/migrate.conf \
-v > migrate_finance.txt
```

In this example, the target zone name is the same as that of the input classic zone, except its distinguished name is different because it is a child zone of the *global* zone. The `-config` parameter specifies the file that contains bind information, in this cases `~/migrate.conf`. The `-v` option directs verbose output to a text file.

You would then run `admigrate` for the next zone to migrate. For example:

```
/usr/share/centrifydc/adedit/admigrate \
-in “cn=engineering,cn=zones,ou=unix,dc=test,dc=org” \
-z “cn=engineering,cn=global,cn=zones,ou=unix,dc=test,dc=org” \
-hz “cn=global,cn=zones,ou=unix,dc=test,dc=org” \
-config ~/admigrate.txt \
-f -v > migrate_eng.txt
```

To simplify the migration process for multiple zones, you could put `admigrate` in a shell script and specify the source zone as an input variable or read it from a file with a listing of all your zones.

Inheritance and overrides

Each time you run `admigrate` with the same parent zone and a different source and target zone, the `admigrate` utility does the following:

- If a user profile from the source zone does not exist in the parent zone, the utility creates a profile for the user in the parent zone.
- If a user profile exists in the parent zone and matches the user profile from the source zone, the new child zone will inherit the user profile attributes as they are defined in the parent zone.
- If a user profile already exists in the parent zone and has attribute values that differ from those for the user from the source zone, the utility creates a user profile in the child zone with overrides for the differing attribute values. For example, if a user profile exists for oscar.romero in the parent zone, but has a different numeric identifier (UID) in the `engineering` zone, the UID attribute value would be different in the `engineering` child zones. The other attributes would be inherited from the parent zone.
- Copies the groups, rights, role definitions, role assignments, and NIS maps from the source zone to the target child zone.

The `admigrate` utility does not copy delegated permissions from the existing classic zones to the new child zones. In addition, delegated permissions are *not* automatically inherited from parent zones to the child zones. After migrating classic zones, you must explicitly delegate administrative permissions on a zone-by-zone basis.

Roles and rights for migrated users

The `admigrate` utility adds the following role definitions for migrated users:

- **login_at_roles** assigns the UNIX system rights **Password login...** and **Non-password login**. It does not assign **Login with non-Restricted Shell** because the user may be assigned to a restricted shell.
- **login_all_apps** assigns the login-all PAM right, which grants access to all PAM applications. It does not assign any UNIX system rights.

By default, all users are added to the **login_all_apps** role so that if they are granted login rights, they have access to all PAM applications, which is the default for users in classic zones. If PAM access rights are restricted by another role assignment, the restricted role assignment will override the rights granted by login_all_apps.

Access uses the following role-assignment rules when migrating roles and rights from a classic zone to a hierarchical zone:

Classic zone	Enabled or disabled	Role assignment in hierarchical child zone
User assigned to role	Enabled	Assign to the following roles: **login_at_roles**, which grants **Password login and Non-password login** UNIX system rights. **login_all_apps**, which grants access to all PAM applications. Corresponding user-created roles, which are migrated.
User assigned to role	Disabled	Assign to corresponding user-created roles, which are migrated. No login roles are assigned because the user is disabled in the classic zone.
User not assigned to role	Enabled	Assign to the default **UNIX Login** role, which grants all UNIX system login rights and access to all PAM applications.
User not assigned to role	Disabled	Assign to the default **listed** role, which makes the user visible in the zone but does not assign any UNIX system rights or PAM access rights.

In classic zones, users who are added to a zone are enabled for login access by default. As an administrator, you can leave a user profile defined in a zone but disable login access.

All the roles and rights you defined in the source zone, as well as any role assignments to user-created roles, are added, as-is, to the child zone each time you run `admigrate`. For example, if you defined a privileged `mount` command in 20 classic zones, `admigrate` will copy that `mount` command to 20 new hierarchical zones. Therefore, after migration you should analyze your role definitions and access right definitions to see if some of them can be moved up to the parent zone to take advantage of inheritance.

Assigning the audit level when migrating

In hierarchical zones, role definition can be assigned an auditing level. This setting is not applicable in classic zones.During migration from classic zones to hierarchical zone, the default "Audit if possible" auditing level, is assigned to all migrated role definitions. After you have migrated, you can change the auditing level in any role definition. For more information about changing the auditing level for a role definition, see Changing the audit level for role definitions.

Moving joined computers to hierarchical zones

After you have migrated data from classic zones to new hierarchical zones, you can move the computers to the new zones using the `adchzone` command-line program.

When you use `adchzone` to change the zone for a computer, the command copies the UNIX profile from the old zone to the new zone, deletes computer profile from the old zone, then stops and restarts `adclient` to flush the cache and update the zone information. The advantage of this approach over leaving the old zone (`adleave`) and then joining (`adjoin`) the new zone is that it is very quick and preserves all the join information without you having to specify join options.

For example, run a command similar to the following to move a computer joined to the classic `finance` zone to the new child `finance` zone:

```
/usr/share/centrifydc/adedit/adchzone \
-z “cn=finance,cn=global,cn=zones,ou=unix,dc=acme,dc=com”
-u finance-adm
```

You will be prompted to supply a password for the specified user.

After changing the zone, you can open Access Manager to see the computer in the Computers node of the new zone, or you can run `adinfo` on the computer to verify the new zone information.

What to do up after migration

The `admigrate` utility migrates most zone information automatically. After using the utility, however, you might want to perform the following tasks to complete the migration:

- Delete unnecessary copies of right and role definitions.

  You should analyze the right and role definitions to see how many of them have been copied into multiple zones. Rights and roles that are defined in the parent zone are available for use in all child zones. By moving role and right definitions to the parent zone you simplify your zone structure making it easier to understand the rights and roles that are available for your organization.

- Review provisioning rules.

  In many cases, hierarchical zones simplify automated provisioning by enabling you to define a baseline profile in a parent zone and only override specific attributes when necessary. If you are using automated provisioning, you should check whether you are defining the provisioning rules in parent zones or in child zones.

- Delegate permissions on a zone-by-zone basis.

  Use the Zone Delegation Wizard to delegate administrative tasks and the assign the corresponding permissions to the appropriate users and groups in your new child zones.

Managing account profiles and identity attributes

This chapter describes how to create user and group profiles that grant access to Centrify-managed Linux and UNIX computers and how to manage identity attributes using Access Manager. No matter what type of zone your environment uses—classic, hierarchical, or auto zone—you can use Access Manager to create and maintain profiles for Active Directory users and groups.

If your environment uses hierarchical zones, you can also use Access Manager to create and maintain profiles for local Linux and UNIX users and groups.

For Active Directory users and groups, you can also perform the tasks described here using ADEdit or Windows PowerShell commands and scripts or other tools, such as Active Directory Users and Computers.

For local users and groups, you can also perform the tasks described here using `adedit` and other command line utilities. See Using Centrify commands for administrative tasks for details about using commands to configure local users and groups.

For additional information about planning the migration of an existing user population, see the *Planning and Deployment Guide*.

Creating group profiles

You can create group profiles for Active Directory groups and—in hierarchical zone environments—local groups. A group profile consists of two attributes and a list of group members. The attributes that must be defined for the group profile to be complete are the following:

- A unique numeric identifier (GID).
- A group name.

A group must have a complete profile with all of these attributes defined to be recognized as a valid group in a zone or on a specific computer. These are the same attributes you define locally for Linux and UNIX groups in the `/etc/group` file.

For details about creating profiles for Active Directory groups, see Creating group profiles for Active Directory groups. For details about creating profiles for local Linux and UNIX groups, see Creating, modifying, and deleting group profiles for local groups.

Creating group profiles for Active Directory groups

You can create a group profile for any domain local, global, or universal security groups you have defined in the Active Directory forest. Associating a group profile with an Active Directory group also enables you to take advantage of any nested group membership you have defined and any group policies you have applied to a domain or organizational unit.

Although associating a group profile with an Active Directory group can be convenient, there is no predetermined requirement to create group profiles for Active Directory groups. Creating a group profile does not create profiles for any members of the group. User accounts must be explicitly given their own profiles.

> **Note:** You can automate the provisioning of account profiles through the use of Active Directory groups. For information about configuring your environment for automated provisioning, see the *Planning and Deployment Guide*.

What to do before creating a new Active Directory group profile

Before you can create Active Directory group profiles, you must have created one or more Active Directory security groups, installed Access Manager, and run the Setup Wizard. You should also identify the specific Active Directory groups for which a group profile is required. In most organizations, only a limited number of Active Directory groups require a zone profile. There are no other prerequisites for performing this task.

Rights required for this task

You must have permission to add groups to a zone. Zone administrators can grant this permission through the Zone Delegation Wizard. If the Active

Directory administrator manually sets the permissions, your user account must be a domain user with the following permissions to create group profiles in a zone:

Select this target object	To apply these permissions
Parent container object for the group profile within the zone	On the **Object** tab, select **Allow** to apply the following permission to this object only: ■ Create serviceConnectionPoint objects Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ■ Read objectClass
Group account object in Active Directory For example: `domain/Users/group_name`	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ■ Read groupType ■ Read objectCategory ■ Read objectClass ■ Read objectGUID ■ Read objectSid
Parent container object for the individual zone For example, if you are adding a group to the `Finance` zone: `domain/UNIX/Zones/Finance`	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ■ Read objectGUID ■ Write Description

Who should perform this task

A Windows domain administrator performs this task, depending on your organization’s policies. In most organizations, this task is delegated to a specific user or group with administrative authority in the selected zone.

How often you should perform this task

In most cases, you only create new group profiles infrequently to address changes to your organization.

Steps for completing this task

If you choose to create group profiles for Active Directory groups, you can use Access Manager, Active Directory Users and Computers, the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API.

The following instructions illustrate how to create a new group profile using Access Manager. Examples of scripts that use ADEdit, Windows PowerShell, or the Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a group profile for an Active Directory group using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the Active Directory group.
3. Expand UNIX Data and select Groups, right-click, then click **Create UNIX Group**.
4. Type a search string to locate the Active Directory group for which you want to create a profile, then click **Find Now**.

   For example, type "fin" to display the `Finance Users` and `Finance Admins` groups.
5. Select one or more groups in the results, then click **OK**.
6. Review the default zone profile settings for the group and make changes if needed, then click **OK**.

   You can deselect an attribute to change the default value or to create a partial group profile in the current zone. You can complete the profile by providing a value for an attribute in a child zone of the current zone. For example, if you use the same group name but different numeric identifiers on two set of computers, you can inherit the group name from a parent zone and set the different numeric identifiers in the child zones.

   In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the group's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the group's zone profile.

   If you selected more than one group, review the profile settings for the each group and modify the default settings, if necessary, then click **OK**.

   If you are adding groups with similar names, you might want to modify the default group name to distinguish the groups. For example, if you are adding both the Finance Admins and Finance Users groups to the same zone, you can change the default group name to `finadmin` and `finuser` to make it easier to tell the groups apart. Keep in mind that in some operating environments group names cannot be more than 8 characters and special characters might not be supported.

Creating, modifying, and deleting group profiles for local groups

When you create a local group profile in Access Manager, it is saved in `/etc/group` on each computer in each zone where the profile is defined. You can create local profiles at the zone level (for example, under **Zones >** *Zonename* **> UNIX Data**) and at the computer level (for example, under **Zones >** *Zonename* **> Computers >** *Computername* **> UNIX Data**). Local group profiles that you create at the zone level are available for local and Active Directory users in the zone and child zones to join.

What to do before creating a new local group profile

You should perform the following tasks before creating local group profiles:

- Ensure that local account management is enabled and configured through configuration parameters or group policies. See Enabling and configuring local account management for more information.
- It is suggested that you review the existing group names in `etc/group` on the computers where the local group profile will be implemented so that you do not attempt to create a group profile with a name that is already used. Access Manager performs a name validation check against `etc/group` in the current zone when you create a new local group. If the group name already exists in `etc/group` somewhere in the current zone, you are prompted to provide a different name for the group that you are creating.

Rights required for this task

The rights required to create local group profiles are the same as the rights required to create Active Directory group profiles. See Rights required for this task for details about those rights.

Using partial profiles and child zones to fine tune group attributes

Access Manager allows you to create a partial profile by leaving any of the attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could define the Members attribute in a parent zone, and then override the parent zone attribute settings by defining the Members attribute differently in different child zones.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to create the group profile.

Groups can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone. If a group profile is still partial at the computer level, the profile is ignored by the agent, and it is not added to `/etc/group` on the local computer. Group profiles must contain the attributes listed in Creating group profiles to be complete.

Specifying profile states

The *profile state* lets you control whether a local group account is in place in `etc/group` and is enabled for use locally. When you create a local group account, you specify the initial profile state. You can change the profile state afterwards to control availability of the local group account. A local group account can have one of the following states:

- **Enable**: If the local group profile is complete, it will be installed or updated in `/etc/group` at the next local account refresh interval.
- **Remove from /etc/group**: The group profile will be removed from `etc/group` at the next local account refresh interval.

You can also choose not to define the profile state by deselecting the **State** check box in the Set Local Group Profile dialog. Deselecting the **State** check box results in one of the following scenarios:

- If a local group profile with the same name exists in the parent zone, the state from the parent group profile is inherited.
- If the parent zone does not contain a group profile with the same name, or if a parent group profile exists but does not define the state, the group profile that you are currently defining is considered incomplete.

Roles and local group account visibility

You use role assignments to control whether local users are visible in a zone. A predefined role definition, `local listed`, is available for use with local user and local group profiles. As with the `listed` predefined role, the `local listed` role does not grant any system rights, PAM rights, or command rights. It is a specialized role that can be used when a local user or local group profile must exist for computers in a zone, but no local user or local group access should be granted.

You can optionally define other roles in the zone to grant visibility to local users and local groups.

By default, all local groups having a complete profile are visible in a zone. You do not have to assign a role to a local group to make the local group visible. However, it is often useful to assign a role (such as `local listed`) to a local group so that all local users in the local group inherit the role assignment, and are visible in the zone.

See Creating, modifying, and deleting user profiles for local users for more information about how roles are used to control visibility of local user accounts.

How often Access Manager and local group accounts are synchronized

The `/etc/group` file on local computers is updated periodically based on the information that you define for local group profiles in Access Manager. The `/etc/group` update interval is controlled by the following group policy and configuration parameter:

- **Group Policy: Set refresh interval for access control cache**, located in Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings.
- **Configuration parameter:** `adclient.refresh.interval.dz`, located in the `/etc/centrifydc/centrifydc.conf` configuration file.

The same group policy and parameter control how often the authorization store cache is updated. Local account information is updated immediately after authorization store information is refreshed in the authorization cache.

For more information, see the *Group Policy Guide*, the *Configuration and Tuning Reference Guide*, and Enabling and configuring local account management.

Steps for completing this task

To create a group profile for a local group using Access Manager

1. Open Access Manager.

2. Expand Zones and any parent zones, child zones, or computers required to select the zone or computer to which you want to add the local group.
3. Expand UNIX Data and select Local Groups.

   You can create a new local group in these ways:

   - **By dragging and dropping an existing local group from another location.** Expand zones or computers to the location of the original local group, and drag it to the location of the new local group. The local group is moved to the new location. To copy (instead of move) the original group, press <Ctrl> while you drag the group.
   - **By cutting or copying an existing local group from another location, and then pasting it into the current location.** Expand zones or computers to the zone where the original local group exists, right-click a local group and select **Cut** or **Copy**, return to the zone where you are creating the new local group, right-click, and select **Paste**.
   - **By creating an entirely new local group.** Perform Step 4 through Step 8 of this procedure.
4. In Local Groups, right-click, then click **Create UNIX Group**.
5. Type a name for the new local group and click **OK**.
6. In the Set UNIX Group Profile dialog, select or deselect check boxes to specify which attributes to set. You must specify at least one attribute to be able to save the profile.

   - **GID:** Type a numeric group ID of your choice.
   - **Members:** Click **Add** to launch the Add Members dialog. In a comma-separated list, type the UNIX names of the users who will be in the group.

     Access Manager does not check the validity of the user names that you provide. You should ensure that all of the names that you provide are UNIX names that currently exist.

     Note that the group profile is considered complete even if this attribute has an empty value.
   - **State:** Specify whether the group account is added to, and enabled in, `etc/group`. Possible values are:

     Enable: The group profile will be installed or updated in `/etc/group` at the next local account refresh interval.

Remove from /etc/group: The group profile will be removed from `etc/group` at the next local account refresh interval.

> **Note:** To modify permissions for a local group, you must first create and save the local group as described in this procedure, and then modify permissions as described in Step 4 in the section To modify group profile attributes and permissions for a local group:.

For the profile to be complete, it must contain settings for group name (specified in Step 5 of the procedure To create a group profile for a local group using Access Manager), GID, and state. You can save the profile now even if it is partial, although it will not be implemented in `/etc/group` until you update it in the current zone, or with settings in child zones, so that it is complete, and you set the state to **Enable**. For example, if you use the same group name but different numeric identifiers on two set of computers, you can inherit the group name from a parent zone and set the different numeric identifiers in the child zones.

In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the local group's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the group's zone profile.

7. Review your local group profile settings and click **OK**.

   If the profile is complete, it is added to `/etc/group` at the next local account refresh interval.

8. To optionally assign the `local listed` role to the local group, so that all local users in the local group are visible in the zone:

   a. At the level where you created the local group, right-click **Role Assignments**, and then select **Assign Role**.

   b. In the Select Role dialog, select **local listed** and click **OK**.

   c. In the Assign Role dialog, ensure that **Accounts below** is selected, and click **Add Local Account**.

   d. In the **Add Local Account** dialog, select **Local UNIX Group** in the **Type** field, type the local group name in the **Account** field, and click **OK**.

   e. In the Assign Role dialog **Accounts below** area, highlight the local group account and click **OK**. The local group is now listed as an assignee of the `local listed` role.

To modify group profile attributes and permissions for a local

group:

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to modify.
2. In the Local Groups details pane, right-click the local group to modify and select **Zone Profile**.

   The Properties dialog for the profile is displayed.
3. Modify attribute selections and settings as described in Step 6 in the procedure To create a group profile for a local group using Access Manager. Keep in mind the following considerations when you change attributes.

   If there is no parent profile for the same local group name:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to define a partial profile.

   If a parent profile for the same local group name already exists in a parent zone:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to inherit attribute values from the parent profile.
4. To optionally modify group permissions (such as read, write, create or delete child object, and so on), click **Permissions**. Refer to the “Active Directory permissions required for administrative tasks” chapter in the*Planning and Deployment Guide* for details about using the Permissions dialog to modify zone-level user and group permissions.
5. Review your changes to the local group profile and click **OK**.

   Your changes are applied to the local group profile in `/etc/group` at the next local account refresh interval.

To delete a group profile for a local group from a zone or computer:

> **Note:** This procedure does not remove a local group profile from `/etc/group`. To remove a local group profile from `/etc/group`, perform the procedure described in To remove a group profile for a local group from /etc/group.

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to delete.
2. In the Local Groups details pane, right-click the local group to modify and select **Delete**.
3. At the warning prompt, select **Yes**.

   The local group is deleted from Access Manager. The group profile still exists in `/etc/group`, but it is ignored.

To remove a group profile for a local group from /etc/group

1. In Access Manager, expand UNIX Data for the zone or computer containing the local group that you want to remove from `/etc/group`.
2. Perform one of the following procedures:
   - Right-click a local group, select **Change Profile State**, then select **Remove from /etc/group**.
   - Right-click a local group, select **Zone Profile**, change the value of the **State** field to **Remove from /etc/group**, and click **OK.**

   At the next local account refresh interval, the local group's profile is removed from `/etc/group`.

Delegating control of local group management tasks

You can use the Zone Delegation Wizard and Computer Delegation Wizard as described in the *Planning and Deployment Guide* to delegate control of local group management tasks.

Migrating local group profiles to Active Directory

In most cases, you get more operational benefits by using Active Directory groups to manage UNIX and Linux user accounts than you would get from migrating your local group profiles into Active Directory. For example, by using Active Directory groups to manage both Windows and UNIX users, you can use your existing provisioning and access control policies across multiple platforms and automate the provisioning and de-provisioning of accounts and access rules.

In some cases, however, you might find it useful to migrate some or all of your existing local groups to Active Directory. If you want to move local group profiles into Active Directory, you have the option to import local groups on a zone-by-zone basis. As part of the import process, you can choose to how each local group should be handled. For example, you can:

- Create a new Active Directory group for each imported group.
- Extend an existing Active Directory group to include an imported group.
- Merge an imported group into an existing UNIX group profile.

Making group membership a requirement

On most Linux and UNIX computers, users can only be members of a limited number of groups at once. Because of this limitation, it is useful to be able to change a user's effective group membership to add and remove groups when necessary. You can use the `adsetgroups` command to dynamically manage the set of Active Directory groups that are available to a user account. You also have the option to specify that membership in a specific group is required in a zone. If you specify that a group is required, users who are members of the group cannot remove the required group profile from their currently active set of groups.

To make membership in a specific group profile required:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name for which you want to add a required group.
3. Expand Groups, then select the group name you want to make required.
4. Right-click, then select **Zone Profile** to display the Centrify UNIX Profile for the group.
5. Select the **Users are required to be members of this group** option.
6. Click **Permissions** to set specific permissions for this group, if needed, then click **OK**.

For more information about using the `adsetgroups` command, see the `adsetgroups` man page.

Overriding and modifying group properties

If you are using hierarchical zones, group profile information is inherited from parent zones into any child zones you define. You can override the inherited profile attributes at any time to create a new group profile in a specific child zone or on individual computers, if needed.

You can also modify either the group profile or the Active Directory group properties for any group at any time using the tool of your choice. For example, you can use Access Manager, the Access Module for Windows PowerShell, ADEdit, Active Directory Users and Computers, or the Centrify Windows API to modify the zone profile or Active Directory properties for a selected group.

Creating user profiles

You can create user profiles for Active Directory users and—in hierarchical zone environments—local users. A user profile consists of the attributes required by the name service switch (NSS) facility on Linux and UNIX computers. User attributes that must be defined for the user profile to be complete are the following:

- A user name (the UNIX login name).
- A unique numeric user identifier (UID).
- The user's primary group profile numeric identifier (GID).
- The default home directory for the user.
- The default login shell for the user.
- General information about the user account (GECOS). (This attribute is required for Active Directory user profiles, but not for local user profiles.)

A user must have a complete profile with all of these attributes defined to be recognized as a valid user in a zone or on a specific computer. You can optionally define other attributes that are not required for the user profile to be complete.

These are the same attributes you define locally for Linux and UNIX users in the `/etc/passwd` file.

For details about creating profiles for Active Directory users, see Creating user profiles for Active Directory users. For details about creating profiles for local

Linux and UNIX groups, see Creating, modifying, and deleting user profiles for local users.

Creating user profiles for Active Directory users

You can create a user profile for any domain user you have defined in the Active Directory forest by adding the user to a zone, or by adding the user to a specific computer in a zone. Associating a user profile with an Active Directory user determines how the Active Directory user is identified on Linux and UNIX computers.

> **Note:** You can automate the provisioning of user profiles through the use of Active Directory groups. For information about configuring your environment for automated provisioning, see the *Planning and Deployment Guide*.

What to do before creating a new Active Directory user profile

Before you can create Active Directory user profiles, you must have created one or more Active Directory users, installed Access Manager, and run the Setup Wizard. You should also identify the computers where Active Directory users might require different profile attributes. For example, you might have some Active Directory users that require the default home directory attribute to be set the to `/home` for access to most computers, but require the attribute to be set to `/Users` when they log on to Mac OS X computers.

In most organizations, Active Directory users have one “dominant” profile with consistent attributes across multiple computers, but require “override” settings to some profile attributes on specific computers or groups of computers. Therefore, most user profiles are only added to parent zones and inherited in child zones.

Rights required for this task

You must have permission to add users to a zone. Zone administrators can grant this permission through the Zone Delegation Wizard. If the Active Directory administrator manually sets the permissions, your user account must be a domain user with the following permissions to create user profiles in a zone:

Select this target object	To apply these permissions
Parent container object for the user profile	On the **Object** tab, select **Allow** to apply the following permission to this object only: ■ Create serviceConnectionPoint Objects This permission is required for both standard zones and RFC 2307-compliant zones. For standard zones, you need to apply additional permissions. Click the **Properties** tab and select **serviceConnectionPoint objects** from the object list, then select **Allow** to apply the following properties to this object: ■ Read Name ■ Read name ■ Read displayName
User account object in Active Directory For example: `domain/Users/user_name`	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ■ Read objectCategory ■ Read objectClass ■ Read objectGUID ■ Read objectSid ■ Read userAccountControl
Parent container object for the individual zone For example, if you are adding a user to the `Finance` zone: `domain/UNIX/Zones/Finance`	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ■ Read objectGUID ■ Write Description

Who should perform this task

A Windows domain administrator performs this task, depending on your organization’s policies. In most organizations, this task is delegated to a specific user or group with administrative authority in the selected zone.

How often you should perform this task

In most cases, you create and remove user profiles frequently to address changes to your user population.

Steps for completing this task

The following instructions illustrate one way to create a new user profile using Access Manager. You can also add a user profile and assign a role to an Active Directory user with the Add User wizard. Examples of scripts that use ADEdit, Windows PowerShell, or the Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a user profile for an Active Directory user using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the Active Directory group.

   In most cases, you should add user profiles to a parent zone.
3. Expand UNIX Data and select **Users**, right-click, then click **Add User to Zone**.
4. Type a search string to locate the user account, then click **Find Now**.

   For example, type "qa" to display the `qa-lab`, `qa-hk` and `qaVenice1x` users.
5. Select one or more users in the results, then click **OK**.
6. Review the default zone profile settings for the user and make any changes if needed, then click **OK**.

   You can deselect an attribute to change the default value or to create a partial user profile in the current zone. You can then complete the profile by providing a value for an attribute in a child zone of the current zone. For example, if you use the same login name but different numeric identifiers on two set of computers, you can inherit the login name from a parent zone and set the different numeric identifiers in the child zones.

   In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the user's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the user's zone profile.

   If you selected more than one user, review the profile settings for the each user and modify the default settings, if necessary, then click **OK**.

Changing the default profile attributes

When you add Active Directory users to a zone, Access Manager displays a default new user profile. You can accept or change the default values for any of the profile attributes, as needed. The default attribute values are automatically generated based on a few simple rules and, in most cases, you can accept them as-is. The following table describes how the default values are populated.

This attribute	Has the following default value
Login name	The Active Directory user logon name associated with the Active Directory account.
UID	A unique number automatically generated by an algorithm based on the security identifier (SID) for the Active Directory user.
Primary group	A unique numeric identifier that represents a private primary group and is the same as the user's default UID. Private groups are not stored or managed in Active Directory.
GECOS	A runtime variable that resolves to the Active Directory `displayName` attribute associated with the Active Directory account.
Home directory	A runtime variable that specifies the default home directory when resolved locally on a computer.
Shell	A runtime variable that specifies the default login shell when resolved locally on a computer. To set the user's shell to the default shell defined for this computer in this zone.

Defining partial UNIX profiles

Access Manager allows you to create a partial profile by leaving any of the attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could leave the Shell attribute blank in a parent zone, define it as `/bin/bash` in a child zone, but override it with `/usr/bin/ksh` in a grandchild zone that only contains AIX computers. You could also leave the Home directory attribute blank in a parent zone, then set it to `/home` in one child zone and to `/Users` on an individual Mac OS X computer that joins the child zone.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to add the user profile. Users must have a complete profile in a zone for any role assignments to be effective. Keep in mind, however, that users can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone to allow role assignments in the child zone.

Defining valid login names

User profile login names can consist of letters, numbers, hyphens, underscores, periods and dashes. Some operating environments may have additional restrictions. For example, some operating environments do not support user names that are longer than 8 characters or require that the first character of the user name be alphabetic. Because UNIX user names typically use only lowercase characters, the default user profile name displayed follows this convention. If you modify the default profile name and include uppercase characters, keep in mind that the proper case must be used when entering the user name. For compatibility with Samba, the dollar sign ($) can also be used at the end of the user name. In general, other special characters, such as ! and &, are not supported.

If the Windows logon name includes unsupported special characters, Access Manager replaces them with underscores for the UNIX login name. For example, Access Manager converts a Windows logon name with special characters, such as `qa:user2` into a valid UNIX login name of `qa_user2`.

Identifying a primary group

In most UNIX environments, a user's primary group identifier (GID) is a "private" group that exists solely for that user. The user is not included as a "member" of the private primary group. You can follow this convention by using a UNIX-only "private" group that is not linked to an Active Directory group, which is the default when you create a new user profile.

If you keep the default private primary group, the primary group identifier (GID) setting in the user profile does not affect the user's actual Active Directory group membership in any way, and there's no need to manage primary groups for UNIX users through Active Directory.

In some cases, however, you might want to assign an Active Directory group that has a corresponding group profile as a user's primary group. If you specify an Active Directory group as a user's primary group, keep in mind that you must manage the membership of that group using Active Directory Users and Computers and that if you identify a group with a large number of members—such as Domain Users—it is likely to affect performance.

For more information about defining primary groups for users, see the *Planning and Deployment Guide*.

Creating, modifying, and deleting user profiles for local users

When you create a local user profile in Access Manager, it is saved in `/etc/passwd` on each computer in each zone where the profile is defined. You can create local profiles at the zone level (for example, under **Zones >** *Zonename* **> UNIX Data**) and at the computer level (for example, under **Zones >** *Zonename* **> Computers >** *Computername* **> UNIX Data**).

After you create local user profiles, you perform a separate set of tasks to create and manage local user passwords. For detailed information about local user passwords, see Creating and managing local user passwords.

What to do before creating a new local user profile

You should perform the following tasks before creating local user profiles:

- Ensure that local account management is enabled and configured through configuration parameters or group policies. See Enabling and configuring local account management for more information.

- It is suggested that you review the existing user names in `etc/passwd` on the computers where the local user profile will be implemented so that you do not attempt to create a user profile with a name that is already used. Access Manager performs a name validation check against `etc/passwd` in the current zone when you create a new local user. If the user name already exists in `etc/passwd` somewhere in the current zone, you are prompted to provide a different name for the user that you are creating.

Rights required for this task

The rights required to create local user profiles are the same as the rights required to create Active Directory user profiles. See Rights required for this task for details about those rights.

Using partial profiles and child zones to fine tune user attributes

Access Manager allows you to create a partial profile by leaving some user attributes blank. Partial profiles can be useful for defining a common set of attributes that are used in multiple zones, then defining specific attributes that vary from one child zone to another or that require different settings on specific computers. For example, you could leave the Shell attribute blank in a parent zone, define it as `/bin/bash` in a child zone, but override it with `/usr/bin/ksh` in a grandchild zone that only contains AIX computers.

If you intend to leave an attribute blank, deselect the attribute check box. However, you must provide a value for at least one attribute to create the user profile.

Users can have an incomplete profile in a parent zone as long as any missing attributes are defined in a child zone. If a user profile is still partial at the computer level, the profile is ignored by the agent, and it is not added to `/etc/passwd` on the local computer. User profiles must contain the attributes listed in Creating user profiles to be complete.

Specifying profile states

The *profile state* lets you control whether a local user account is in place in `etc/passwd` and is enabled for use locally. When you create a local user account, you specify the initial profile state. You can change the profile state afterwards to control availability of the local user account. A local user account can have one of the following states:

- **Enable**: If the user profile is complete, it will be installed or updated in `/etc/passwd` at the next local account refresh interval. The user can log into the local computer, and is visible in Access Manager if a role with the visible right (such as `local listed`) is granted to the user. See Roles and local user account visibility for more information about how roles affect local user visibility.
- **Disable**: If the user profile is complete, it will be installed or updated in `/etc/passwd` at the next local account refresh interval. However, the user will not be able to log into the local computer. This state results in what is typically called a “locked account.” UNIX and Linux service accounts and system accounts are typically set up as locked accounts.
- **Remove from /etc/passwd**: The user profile will be removed from `etc/passwd` at the next local account refresh interval.

You can also choose not to define the profile state by deselecting the **State** check box in the Set Local User Profile dialog. Deselecting the **State** check box results in one of the following scenarios:

- If a local user profile with the same name exists in the parent zone, the state from the parent user profile is inherited.

- If the parent zone does not contain a user profile with the same name, or if a parent user profile exists but does not define the state, the user profile that you are currently defining is considered incomplete.

Roles and local user account visibility

You use role assignments to control whether local users are visible in a zone. A predefined role definition, `local listed`, is available for use with local user and local group profiles. As with the `listed` predefined role, the `local listed` role does not grant any system rights, PAM rights, or command rights. It is a specialized role that can be used when a local user profile must exist for computers in a zone, but no local user access should be granted.

You can optionally define other roles in the zone to grant visibility to local users.

As with role assignments for Active Directory users, local user role assignments can be made at the zone level, computer level, or computer role level. Use the following guidelines to establish where local users are visible in Access Manager:

- To make a local user visible to all computers in a zone, assign the `local listed` role to the local user account (or to all local UNIX accounts) in the zone (for example, assign `local listed` to users located in **Zones** > *Zonename* > **UNIX Data** > **Local Users**).
- To make a local user visible only to a specific computer, assign the `local listed` role to the local user account (or to all local UNIX accounts) located in the computer zone (for example, assign `local listed` to users located in **Zones** > *Zonename* > **Computers** > *Computername* > **UNIX Data** > **Local Users**).
- To make a local user visible only to a group of computers, create a computer role and assign the `local listed` role to the local user account (or to all local UNIX accounts) in the computer role.

How often Access Manager and local user accounts are synchronized

The `/etc/passwd` file on local computers is updated periodically based on the information that you define for local user profiles in Access Manager. The `/etc/passwd` update interval is controlled by the following group policy and configuration parameter:

- **Group Policy: Set refresh interval for access control cache**, located in Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings.
- **Configuration parameter:** `adclient.refresh.interval.dz`, located in the `/etc/centrifydc/centrifydc.conf` configuration file.

These are the same group policy and parameter that control how often the authorization store cache is updated. Local account information is updated immediately after authorization store information is refreshed in the authorization cache.

For more information, see Enabling and configuring local account management of this guide. For additional group policy and configuration parameter information, see the *Group Policy Guide*, and the *Configuration and Tuning Reference Guide*.

Steps for completing this task

To create a user profile for a local user using Access Manager, Method 1

> **Note:** This method begins from the Local Users node, and allows you to assign just one role, `local listed`, to the local user. To use the Add User to Zone wizard, which lets you assign other roles to the local user, see To create a user profile for a local user using Access Manager, Method 2.

1. Open Access Manager.
2. Expand Zones and any parent zones, child zones, or computers required to select the zone or computer to which you want to add the local user.
3. Expand UNIX Data and select Local Users.

   You can create a new local user in these ways:

   - **By dragging and dropping an existing local user from another location.** Expand zones or computers to the location of the original local user, and drag it to the location of the new local user. The local user is moved to the new location, and no longer exists in the original location. To copy the original user to the new location and also retain it in the original location, press <Ctrl> while you drag the user.
   - **By cutting or copying an existing local user from another location, and then pasting it into the current location.** Expand zones or

computers to the zone where the original local user exists, right-click a local user and select **Cut** or **Copy**, return to the zone where you are creating the new local user, right-click, and select **Paste**.

- **By creating an entirely new local user.** Perform Step 4 through Step 8 of this procedure.

4. In Local Users, right-click, then click **Add User to Zone**.
5. Type a name for the new local user and click **OK**.
6. In the Set UNIX User Profile dialog, select or deselect check boxes to specify which attributes to set. You must specify at least one attribute to be able to save the profile.

   If a parent profile for the same local user name already exists in a parent zone, some attribute fields will be filled in already with inherited values. You can edit profile fields to customize inherited values, and you can deselect other profile fields to inherit attribute values from the parent profile.

   - **UID:** Type a numeric user ID of your choice.
   - **Primary group:** From the drop-down list, select an existing group, or select **<Not defined>** to leave the PGID attribute undefined, or select **<...>** to see additional group choices or to create a new local group.

     To create a new local group after clicking **<...>**, click **Add** in the Select a Group dialog, and follow the procedure for creating a new local group starting with Step 5 in the section To create a group profile for a local group using Access Manager.
   - **GECOS:** Optionally type general information of your choice about the local user account. This attribute is not required for the profile to be complete.
   - **Home directory**: Type the default local computer home directory for the local user.
   - **Shell**: Select the default shell for the local user. Choices are `/bin/bash`, `/bin/csh`, `/bin/ksh`, `/bin/sh`, `/bin/tcsh`, `%{shell}`.
   - **State:** Specify whether the local user account is added and enabled in `/etc/passwd`. Choices are as follows.
     - **Enable**: If the user profile is complete, it will be installed or updated in `/etc/passwd` at the next local account refresh interval. The user will be able to log into the local computer,

and the user is visible in Access Manager.

- **Disable**: If the user profile is complete, it will be installed or updated in `/etc/passwd` at the next local account refresh interval. However, the password field in `/etc/passwd` will be set to `!!`, and the user will not be able to log into the local computer. This state results in what is typically called a “locked account.” The user is still visible in the zone as long as the `local listed` role is assigned to the user.
- **Remove from /etc/passwd**: The user profile will be removed from `etc/passwd` at the next local account refresh interval.

For the profile to be complete, it must contain the attributes listed in Creating user profiles. You can save the profile even if it is partial, although it will not be implemented in `/etc/passwd` until you update it in the current zone, or with settings in child zones, so that it is complete, and you set the state to enabled. For example, if you use the same user name but different numeric identifiers on two set of computers, you can inherit the user name from a parent zone and set the different numeric identifiers in the child zones.

In the **AIX Extended Attributes** tab, you can view and set AIX attributes for the local user's zone profile. Click **Add** to add an attribute and a value, click **Edit** to change an attribute, or click **Remove** to remove an attribute from the user's zone profile.

Note: To modify permissions for a local user, you must first create and save the local user as described in this procedure, and then modify permissions as described in To modify user profile attributes and permissions for a local user:.

7. By default, new local users are assigned the `local listed` role so that local users are visible in Access Manager. This assignment is specified in the **Assign local listed role to make this user visible** check box. To keep this default assignment, ensure that the check box remains selected.

   To give the local user a different role assignment, deselect the check box. If you deselect the check box, you will need to manually assign a role with visible rights to the local user after completing this procedure.

8. Review your attribute selections and settings, and click **OK**. If the user profile is complete, it is added to `/etc/passwd` at the next local account refresh interval.

To create a user profile for a local user using Access

Manager, Method 2

Note: This method describes how to create a local user profile using the Add User to Zone wizard, which lets you assign roles other than just `local listed` to the local user.

1. Open Access Manager.
2. Expand Zones and any parent zones, child zones, or computers required to select the zone to which you want to add the local user.
3. Right-click the zone, and select **Add User**.

   The Add User to Zone wizard launches.
4. In the Select User Type dialog, select **Local UNIX user**, and click **Next**.
5. In the Specify Local UNIX User dialog, type a name for the local user, and click **Next**.
6. In the Add User to Zone dialog, select the **Define user UNIX profile** and **Assign roles** check boxes. Click **Next**.
7. Fill in the local users profile attribute settings in the Define User UNIX Profile dialog as described in Step 6 in the section To create a user profile for a local user using Access Manager, Method 1, and click **Next**.
8. In the Assign Roles dialog, the `local listed` role is included by default. To optionally add different roles as choices, click **Add** and select one or more roles to add to the list.
9. In The Assign Roles dialog, select one or more roles, and click **Next**.
10. In the Confirm Your Selections dialog, review your choices and click **Next**.
11. In the final wizard screen, click **Finish**.
12. Confirm that the new local user was created by expanding UNIX Data in the zone and clicking **Local Users**. The new local user should be listed in the user details pane.

To modify user profile attributes and permissions for a local user:

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to modify.
2. In the Local User details pane, right-click the local user to modify and select **Zone Profile**.

The Properties dialog for the profile is displayed.

3. Modify attribute selections and settings as described in Step 6 in the section To create a user profile for a local user using Access Manager, Method 1. Keep in mind the following considerations when you change attributes.

   If there is no parent profile for the same local user name:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to define a partial profile.

   If a parent profile for the same local user name already exists in a parent zone:

   - You can edit profile fields to customize the value.
   - You can deselect profile fields to inherit attribute values from the parent profile.

4. To optionally modify user permissions (such as read, write, create or delete child object, and so on), click **Permissions**. Refer to the “Active Directory permissions required for administrative tasks” chapter in the *Planning and Deployment Guide* for details about using the Permissions dialog to modify zone-level user and group permissions.

5. Review your changes to the local user profile and click **OK**.

   Your changes are applied to the local user profile in `/etc/passwd` at the next local account refresh interval.

To disable a user profile for a local user:

Note: This procedure does not remove a local user profile from `/etc/passwd`. To remove a local user profile from `/etc/passwd`, perform the procedure described in “To remove a user profile for a local user from /etc/passwd.”

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to disable.

2. In the Local Users details pane, right-click the local user and select **Change Profile State**.

3. Select **Disable**.

   The local user remains visible in Access Manager. At the next local account refresh interval, the local user’s profile in `/etc/passwd` is modified so that the password field contains `!!`, and the user cannot log into the local computer.

To delete a local user from a zone

> **Note:** This procedure does not remove a local user profile from `/etc/passwd`. To remove a local user profile from `/etc/passwd`, perform the procedure described in “To remove a user profile for a local user from /etc/passwd before you delete the local user from the zone.

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to delete from the zone.
2. In the Local Users details pane, right-click the local user and select **Delete**.
3. In the confirmation dialog, select **Yes** to delete the user from the zone. To prevent the confirmation dialog from displaying in the future, select **Do not warn me again**.
4. In the next confirmation dialog, select **Yes**.

   The user is removed from zone, and is no longer controlled through Access Manager. However, the user profile remains in `/etc/passwd` on local computers.

To remove a user profile for a local user from /etc/passwd

1. In Access Manager, expand UNIX Data for the zone or computer containing the local user that you want to remove.
2. In the Local Users details pane, right-click the local user and select **Change Profile State**.
3. Perform one of the following procedures:
   - Right-click the local user, select **Change Profile State**, then select **Remove from /etc/passwd**.
   - Right-click the local user, select **Zone Profile**, change the value of the **State** field to **Remove from /etc/passwd**, and click **OK.**

At the next local account refresh interval, the local user’s profile is removed from `/etc/passwd`.

Delegating control of local user management tasks

You can use the Zone Delegation Wizard and Computer Delegation Wizard as described in the *Planning and Deployment Guide* to delegate control of local user management tasks.

Creating and managing local user passwords

After you create local user profiles as described in the preceding sections, you still need to assign a password to each user. You can create local user passwords in one of these ways:

- By creating a shell script to execute the `passwd` command on each local computer, giving each local user the password that you specify in the script. The shell script can be executed manually, or by enabling `adclient.local.account.notification.cli` to run the script automatically when local accounts are refreshed. This is the least secure way to assign passwords to local users, because the same password is assigned to each user when the script runs. After the script runs, you must change passwords locally so that each password is unique.

  This guide does not include detailed instructions for implementing this method of creating local user passwords.

- If your environment contains a third-party password management product, you can create a shell script that executes on each local computer, giving each local user a random password. The shell script can include a section that submits the passwords to the password management product for storage and maintenance. The shell script can be executed manually, or by enabling `adclient.local.account.notification.cli` to run the script automatically when local accounts are refreshed.

  A sample shell script, `handle_local_accts.sh`, is provided in `/usr/share/centrifydc/samples/localacctmgmt` for you to use as a reference when you create your own shell script. Typically, the shell script that you create should perform the following tasks:

  - Assign a random password to newly provisioned local users, and to local users whose accounts were recently unlocked (that is, re-enabled after having been disabled).

  - Optionally create a home directory for each new local user.

  - Provide the user account information, including the generated passwords, to a third-party password management solution.

  For syntax details about the notification CLI, execute the sample script with the `-h` option:

  ```
  handle_local_accts.sh -h
  ```

- If your environment does not contain a third-party password management product and you want to create and maintain unique passwords for each local user, you can use Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service to manage local user passwords.

  Using Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service to manage local user passwords involves these tasks:

  - Register for Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service.
  - Download the Centrify agent for Linux software package.
  - On each UNIX and Linux computer where you will assign passwords to local users, execute the `cenroll` command to register the computer as a managed resource.
  - Create a shell script that executes on each local computer, giving each local user a random password. The shell script should include commands to manage generated passwords. The agent package includes a sample shell script that you can use as a reference when you create your own shell script.
  - Enable the `adclient.local.account.notification.cli` configuration parameter to run the shell script automatically when local accounts are refreshed.

Setting runtime variables in user profiles

Access Manager maintains a set of predefined runtime variables that you can use in place of specific values in Active Directory user profiles and local user profiles. Using the variables simplifies the process of defining profile attributes. The Centrify UNIX agent resolves the runtime variables defined in a profile with appropriate values when a computer joins a domain and zone.

The predefined runtime variables you can use in profiles are:

Use this variable	To specify this
%{domain}	The domain to which the computer is joined.
%{home}	The root home directory. By default, this directory is `/home` on most Linux and UNIX computers. For Mac OS X computers, the default home directory is `/Users`. On Solaris computers, the default home directory is `/export/home`).
%{host}	The host name of the joined computer.
%{shell}	The default login shell for the user. By default, the shell is `/bin/bash` on most Linux and UNIX computers. On Solaris and HP computers, the default shell is `/bin/sh`. On AIX computers, the default shell is `/usr/bin/ksh`.
%{site}	The Active Directory site of the joined computer.
%{user}	The user's UNIX login name. **Note:** This variable is supported only for Active Directory users. It is not supported for local users.
%{zone}	The zone to which the computer is joined.

You can use these predefined runtime variables or custom variables at any point in the zone hierarchy, including a parent zone, a child zone, or on individual computers. At runtime, the `adclient` process resolves the variables based on how the following configuration parameters are set and where the variables are defined in the zone hierarchy:

- `nss.runtime.defaultvalue.var.`variableName

  These parameters — one for each predefined variable — defines the default value for each parameter as shown in the table. These are the values are used if the variable is not explicitly defined in the zone or by the `nss.runtime.var.`variableName parameter in the configuration file. For example:

  ```
  nss.runtime.defaultvalue.var.home: /home
  nss.runtime.defaultvalue.var.shell: /bin/bash
  ```

- `nss.runtime.var.`variableName

  These parameters allow you to specify a specific value for any of the predefined variables in the configuration file. The value in the configuration file is essentially a computer-specific override because it applies only to the computer on which it is defined and overrides any other setting for the variable, including the default value, or a specific value in a zone Properties page. For example:

  ```
  nss.runtime.var.home: /Users
  nss.runtime.var.shell: /bin/sh
  ```

To override the default definition for any predefined variable in a zone, you can simply add a variable with the same name to the zone by using the zone Properties page or by using ADEdit. Zone variables and zone variable definitions are inherited down the profile tree, which means that a variable could have one definition at the top of the tree and a different definition at the bottom. The value that is applied depends at which level of the zone hierarchy a computer joins the domain.

To define values for predefined variables in a parent or child zone:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name in which you want to override a profile attribute.

   For example, if you want to override the default login shell in the child zone that only AIX computers join, you might expand Child Zones to view and select the `IBM AIX Only` zone.
3. Select the zone, right-click, then click **Properties**.
4. Click the **Variables** tab, then click **Add**.
5. Type the name of the predefined variable and the custom value you want to use, then click **OK** to save the variable definition.

   For example, type `shell` and set the value to `/usr/bin/ksh` to modify the default shell definition.
6. Click **OK** to close the zone properties.

Using Active Directory attributes as variables

You can also use any Active Directory user attributes as variables by specifying the attribute name in the following format:

```
%{u:attributeName}
```

For example, if you want to populate the GECOS field of a user's zone profile with the information from the user's `department` attribute, you could specify the variable as follows:

```
%{u:department}
```

By default, only a subset of common user object attributes can be retrieved and resolved by the adclient process. The default set of attributes you can use in a user profile are:

- mail
- department
- description
- mobile
- title
- telephoneNumber

The most common format for the GECOS field in a user profile contains the user's full name, building number, and office phone number separated by commas. Depending on the operating system and desktop manager you are using, the information from the GECOS field might also be used to display the user name when logging on. If you specify an attribute for the GECOS field that includes a comma, you might see the first part of the attribute treated as the user's full name and displayed in the login screen. For example, if you are using the department attribute in the GECOS field and the attribute is defined as "Cendura, San Francisco, Engineering, 25th floor, office 202", you might see Cendura listed as a user on the login screen.

Using other attributes in a profile

The default user attributes are recognized by adclient without requiring any modification to the managed computer or Active Directory. If you want to use any other attribute, whether it is a standard schema attribute like company or homePhone or a custom attribute that you have added to the Active Directory schema such as supervisorId, you must add an entry for the attribute to the adclient.custom.attributes.user parameter in centrifydc.conf file, then restart adclient and flush the cache.

For example, you might add the following attributes to the centrifydc.conf file:

```
adclient.custom.attributes.user: company supervisorId
```

After modifying the file, you would run the following commands to restart the agent and clear the cache:

```
/usr/share/centrifydc/bin/centrifydc restart
adflush -f
```

For more information about defining custom attributes, see the *Configuration and Tuning Reference Guide*.

Attributes for users in a forest with a one-way trust

Keep in mind when using attribute variables that if you add users to a zone from a one-way trusted forest, the Centrify agent will only be able to retrieve values for the `userPrincipalName` and `samAccountName` attributes. Therefore, at runtime, when the `adclient` process resolves variable definitions, fields that contain any other variables will be blank for a user from a one-way trusted forest.

Adding custom variables to a zone

You can also create your own variables at any point in the zone hierarchy, including a parent zone, a child zone, or on individual computers. You can add custom variable names and values in exactly the same way you define new values for the predefined runtime variables, except that you type a custom variable name and value.

Importing local account profiles

Most organizations have at least some local user and group profiles that must be migrated to Active Directory. Access Manager provides an Import from UNIX wizard that enables you to import user and group profiles from local `/etc/passwd` and `/etc/group` files or from NIS servers and domains.

If you are not migrating any local account profiles, you can skip this section. However, if you have a large or complex user population to migrate, you should use the information in this section along with the *Planning and Deployment Guide* for a more complete view of the migration process and analysis requirement.

Collecting account information

Before using the Import from UNIX wizard, you should do the following to prepare:

- Identify each source of user information and analyze the information to determine your zone requirements.
- Run appropriate commands—such as `getent passwd`, `getent group`, or `niscat`—to export user and group information and save it in properly-formatted text files.

  Copy the text files to a location that is accessible from the Windows network. If you want to import information directly from NIS maps instead of text files, you should verify that you can access NIS servers and domains from the Windows network.
- Review the text files entries to remove account entries that don't need to be mapped to Active Directory accounts.

  You can automatically exclude system accounts with UID or GID values from 0 to 99 during the import process, but you might want to remove other accounts prior to the import. As part of the review process, determine which entries should map to existing Active Directory accounts or which entries require new Active Directory objects.

Using variables when importing UNIX users

When you import UNIX user accounts, you can use a variable in the GECOS field so that Active Directory will automatically populate that information. The variable you can use is as follows:

```
% {u:xxx}
```

For example: In your /etc/passwd file, you have the following information for a user:

```
ron:x:10061:10061:%{u:displayName}:/home/ron:/bin/bash
```

After you import the user with the UNIX import user wizard, the following user is in the pending import area:

```
UID: 10061
Login name: ron
Shell: /bin/bash
Home directory: /home/ron
Primary Group: 10061
GECOS: %{u:displayName}
```

After you map this pending user to a user account in Active Directory, the `%{u:displayName}` text is converted to the user's display name at runtime by adclient. When you view the user profile in Active Directory or Access Manager, you'll see the `%{u:displayName}` text in the GECOS field; when you query the user from a UNIX computer using something such as adquery or getpwent, you'll see the actual user display name in the GECOS field.

Using the Import from UNIX wizard

After you have created text files with user and group information or verified access to a NIS server and domain, you are ready to perform the first step in the migration process using the Import from UNIX wizard.

To import user and group information:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name into which you want to import users and groups.
3. Select UNIX Data, right-click, then click **Import from UNIX**.
4. Select the import source, then click **Next**.
   - If you select Network Information Service (NIS), type the name of the NIS domain and the host name of the NIS server. The NIS domain and server must be accessible from the Windows network for information to be imported successfully.
   - If you select UNIX configuration files, click **Browse** to locate the text files to import.

   If you selected **Network Information Service** or **UNIX configuration files** in Step 4, go to Step 5.
5. Select the import options you want to use, then click **Next**.

   The import options displayed depend on the import source. For example, if you selected UNIX configuration files and specified a text file containing user accounts and a text file containing group accounts, the import options are:

- **Include system accounts** to include accounts with UID or GID values from 0 to 99.

  On most computers, accounts with UID or GID values from 0 to 99 are reserved for accounts, such as `root`, `tty`, and `ftp` that you don't need to import or manage using Active Directory. Select the **Include system accounts** option to include these accounts. This option is only displayed if importing from UNIX configuration files.

- **Automatically shorten the UNIX name to 8 characters** to limit UNIX user and group names to a maximum of 8 characters.

  On some computers, user and group names cannot be longer than 8 characters. If you are importing users and groups that might need access to computers that do not support names longer than 8 characters, you can select **Automatically shorten the Unix name to 8 characters** to automatically truncate the names imported.

If you are importing from NIS, you can choose to import users, groups, or both.

6. Select a location for storing pending import data, then click **Next**.

   For example, to store pending data for the current zone in an XML file, select **Store in XML file** and specify the location for the file. If the file does not already exist in the default location, you are prompted to create it. To select another location for the XML file, click **Browse**.

7. Review the summary of information to be imported, and select the **Check data conflicts while importing** option if you want to check for conflicts and potential matching candidates during the import process, then click **Finish**.

   If you are importing a large number of users or groups, selecting **Check data conflicts while importing** can cause the import process to take some time to complete. If you don't select this option, you must check the status of users or groups after importing.

After you close the Import from UNIX wizard, users and groups are placed in Active Directory or in an XML file with the status of Pending Import. You must then decide how each user and group should be mapped to accounts in Active Directory.

Checking for conflicts and matching candidates

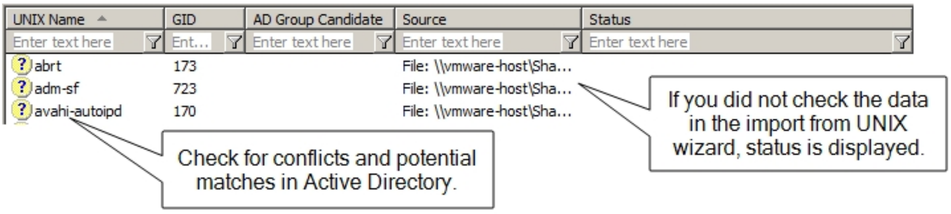
To move a user or group from Pending Import to a UNIX profile attached to an Active Directory user or group, you must first check for potential conflicts and for potential matching user or group candidates in Active Directory. If you selected the **Check data conflicts while importing** option in the Import from UNIX wizard, you have already completed this step and can continue to Mapping UNIX profiles to Active Directory accounts.

To check the status of pending information:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name into which you imported users and groups.
3. Expand UNIX Data, then expand Groups and Users to see the Pending Import nodes.

   For example, if you imported information for the "Finance" zone, open that zone, expand UNIX Data, then expand Groups and Users.
4. Select Pending Import to display the list of users or groups to be imported.

   For example:



UNIX Name	GID	AD Group Candidate	Source	Status
Enter text here	Ent...	Enter text here	Enter text here	Enter text here
? abrt	173		File: \\vmware-host\Sha...	
? adm-sf	723		File: \\vmware-host\Sha...	
? avahi-autoipd	170		File: \\vmware-host\Sha...	

Check for conflicts and potential matches in Active Directory.

If you did not check the data in the import from UNIX wizard, status is displayed.

5. Select all or a subset of pending import users or groups, right-click, then click **Check status**.
   - For pending import groups, a potential match is an Active Directory group with a common name or `sAMAccountName` that is the same as the pending import group name.
   - For pending import users, a potential match is an Active Directory user with a common name that is the same as the pending import user's GECOS field, or `sAMAccountName` that is the same as the UNIX user name.

   If there is a match, Access Manager displays that group or user as the default Active Directory candidate and the status as **Ready to import**.

If Access Manager can't identify a potential match in Active Directory or there are other issues, the status for the pending import group or user describes the issue encountered.

Mapping UNIX profiles to Active Directory accounts

After you check the status of pending import groups or users, you can map the pending import group or user to an Active Directory group or user. The actions you can take depend on the object you select and its current state. For example, if you select a pending group, you can choose to:

- Accept the default Active Directory candidate for the selected group if a candidate is identified.
- Create a new Active Directory group and attach the selected UNIX group profile to it.
- Extend an existing Active Directory group to include the selected UNIX group profile.
- Merge the members of the selected UNIX group with an existing UNIX group in Active Directory.
- Delete the selected UNIX group.
- View and modify the properties of the selected UNIX group.

Accepting the Active Directory candidate

If Access Manager finds a potential match for the pending import group or user in Active Directory, it displays the matching candidate in the details pane. You can accept the suggested candidate by right-clicking the pending import group or user, then selecting **Accept**. After you accept the Active Directory candidate for a pending group or user, the group or user is removed from the Pending Import list.

If all of the pending import group members have an Active Directory candidate associated with them, they are added as members of the Active Directory group. However, the group will remain in the Pending Import list until all of its members are successfully mapped to Active Directory users or removed as members.

Creating a new Active Directory account

If Access Manager did not find a potential match in Active Directory, you must determine whether the pending import group or user should be mapped to an existing Active Directory account or requires a new Active Directory account. If the pending group or user requires a new Active Directory account, right-click the pending group or user, then select the **Create new** option to open the wizard for creating a new Active Directory group or a new Active Directory user.

Follow the prompts displayed in the wizard to provide the additional information needed to create the group or user account.

Adding a profile to an existing Active Directory account

If Access Manager did not find a potential match in Active Directory but an appropriate Active Directory account exists, you must map the pending import group or user to the appropriate Active Directory group or user. If the pending import profile should be added to an existing Active Directory group or user, right-click the pending group or user, then select the **Extend existing** option to open the wizard for adding a UNIX profile to an existing Active Directory group or existing Active Directory user.

Merging pending group members into an existing group

If Access Manager did not find a potential match for a Pending Import group in Active Directory, you might want to merge the members of the Pending Import group into a group that already has a UNIX profile in the zone. If you want to add the members of a selected pending import group to an existing group profile, right-click the pending import group, then select the **Merge into existing Unix group** option to open the wizard for merging the membership of a pending import group with the membership of an existing UNIX group.

Deleting a UNIX profile for a pending group or user

If there are no suitable candidates to map a pending import group or user, you might want to remove a pending group or user from the Pending Import list. If you want to delete a pending import group or user, you can do so by right-clicking the pending import group or user, then selecting the **Delete** option.

Viewing or modifying properties for a pending group or user

If there are conflicts between a pending import profile and information in Active Directory, you might need to modify the properties associated with the pending import profile before you can take any other action. If you want to view or modify the properties for a pending import group or user, right-click the pending import group or user, then select **Properties**.

If you select a pending group, the properties include the UNIX profile, the time of the import, the file location the information was imported from, the members of the group, and the status of the group.

If you select a pending user, the properties include the UNIX profile, the time of the import, the file location the information was imported from, and the status of the user.

Resolving errors and conflicts

In some cases, you might encounter errors (❌) that must be resolved before a pending import user or group can be migrated into Active Directory. For example, pending import groups cannot be imported if the group profile has any of the following problems:

- The group’s GID is negative.
- There is another UNIX group with the same GID already defined in the zone.
- There is a UNIX group with the same group name already defined in the zone.
- The matching Active Directory candidate already has a UNIX profile in the zone.

Similarly, pending import users cannot be imported if the user profile has any of the following problems:

- The user’s UID is negative.
- The user’s primary group GID is negative.
- There is a UNIX user with the same user name already defined in the zone.

In most cases, you must resolve these issues by modifying the properties for the pending import profile. For example, assume you are importing a passwd file that includes the UNIX user account `pierre` with the UID 1001, but there is already an UNIX profile in the zone with the UNIX name `pierre` and UID of 500. After you check the status, the Pending Import list of users will indicate there is an error.

To resolve a conflict like this, you might select the pending import user, right-click, then select **Properties** to change the UNIX user name from `pierre` to another name, such as `pierre2`. You should keep in mind, however, that conflicts like this might require investigation to determine the appropriate course of action. For example, if you are attempting to import the UNIX profile for the user `pierre` and there's a conflict, you need to determine whether `pierre` with the UID of 1001 is the same person as `pierre` with a UID of 500 and where each UID is applicable. If both profiles are for one person accessing different computers, you might simply need to define a computer-level override on the specific computer where the UID of 1001 is required. If the pending import user actually refers to a different person, you might have to map the profile to a different Active Directory account or move the computer to a different zone.

Resolving warnings

In addition to the errors that prevent users or groups from being imported, there are several conditions that generate a warning (⚠). Warnings indicate potential problems that you should try to resolve. After you check the status for pending import groups and users, the most common warning is “No matching Active Directory candidate is found.” To continue, you must identify or create an Active Directory account for the pending import profile.

If you make changes to a pending import user or group to correct problems, you should click **Check status** after the change to check for any additional issues that might need to be resolved.

Overriding and modifying user properties

If you are using hierarchical zones, user profile information is inherited from parent zones into any child zones you define. You can override the inherited profile attributes at any time to create a new user profile in a specific child zone or on individual computers, if needed. Overriding profile attributes enables you to migrate legacy local accounts without modifying any existing account information or file and directory ownership.

You can also modify either the user profile or the Active Directory user account properties for any user at any time using the tool of your choice. For example, you can use Access Manager, the Access Module for Windows PowerShell, ADEdit, Active Directory Users and Computers, or the Centrify Windows API to modify the zone profile or Active Directory properties for a selected user.

To override a profile attribute in a user profile:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name in which you want to override a profile attribute.

   For example, if you want to override the default login shell in the child zone that only AIX computers join, you might expand Child Zones to view and select the `IBM AIX Only` zone.

   If you want to override a profile attribute for a specific computer, expand Computers to select the computer name on which you want to override the profile attribute. For example, if you want to override the default numeric identifier for a user on the AIX computer `aix6v0.ajax.org`, you might expand the `IBM AIX Only` child zone and the Computers node to view and select the `aix6v0.ajax.org` computer.
3. Expand UNIX Data for the zone or computer, then select Users.
4. Select the user, right-click, then select **Zone Profile**.

   The profile displays the attributes inherited from the parent zone or currently set.
5. Select an attribute and provide the override value.

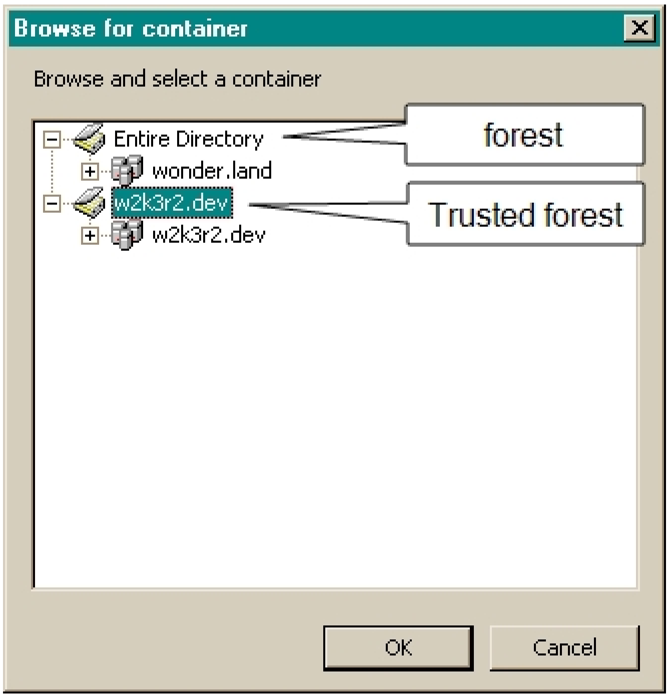
   For example, select **Shell** and type `/usr/bin/ksh` to give the selected user profilea different default login shell—one appropriate for an AIX computer—in the selected zone or on the selected computer.
6. Click **OK** to save the profile change.

Adding users or groups from a trusted forest

In most cases, when you create a profile for a user or group in a zone, the Active Directory account already exists in the local Active Directory forest. You can, however, also add profiles for remote users and groups to a zone without adding them to the local forest. If you have established a one-way or two-way trust relationship with a remote or external Active Directory forest, you can add users and groups from that forest to a selected Centrify zone.

You add remote or external users and groups to the zone in the same way you add profiles for local Active Directory users and groups except that you must select the remote forest or domain before searching for the user or group account. For example, at Step 4 of the procedure To create a group profile for an Active Directory group using Access Manager:, click **Browse** to select a trusted external forest or a specific domain in the trusted forest.

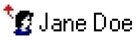
If you have defined a one-way or two-way trust between a local forest (`wonder.land`) and a remote forest (`w2k3r2.dev`), you can select the remote forest in the Browse for container dialog box to add groups from that forest (`w2k3r2.dev`) to the currently selected zone.



If you use attribute variables to define any part of the user profile, keep in mind that the Centrify agent cannot directly read any of the attributes for a user from a one-way trusted forest. The agent can retrieve the `userPrincipalName` and `sAMAccountName` from the zone profile for the user. However, the agent cannot retrieve other user attributes. If the agent cannot resolve a variable in the user profile, the agent leaves the attribute value undefined. For example, if you use the `displayName` variable to define the GECOS attribute, that attribute will be undefined for all users from an external forest with a one-way trust.

Identifying users from remote forests

You can identify the Active Directory users who have been added from a remote or external forest by checking the icon displayed in the Access Manager console. If a user is added from a remote or external forest, the user name displays the following icon:



Jane Doe

Valid login names for users from a remote forest

If you add users from an external forest to a zone, you should be aware that those users can only log on or be identified using the following information:

- A valid UNIX profile name that has a complete set of profile attributes.
- The full Active Directory user name including the user's external forest domain name.

When users are defined in a local forest, they can be located in Active Directory by their UNIX profile name, their `userPrincipalName`, or their `sAMAccountName` in the form of their user logon name alone or in the format of domainname\username, so any of these login name formats can be used to access user information or to log on to a Centrify-managed computer.

To identify a user from a trusted external forest, however, you must use either the user's UNIX profile name for the zone or the user's `sAMAccountName` followed by the user's external domain name in the form of sAMAccountName@domainname. Using the UNIX profile name or the sAMAccountName@domainname ensures the name is unique when there are cross-forest trust relationships. For example, if an Active Directory user from a trusted external forest (`sierra.org`) has the Active Directory logon name of `sofia.perez` and a UNIX profile name of `sofiapz`, the user can be identified using:

- `sofia.perez@sierra.org`
- `sofiapz`

You cannot use `sierra\sofia.perez` or `sofia.perez` without the domain to retrieve information or authenticate from a remote forest. In addition, the `userPrincipalName` (username@domainname) for any user might be different from the sAMAccountName@domainname. For example, if you use alternate UPN suffixes, the domain name used in the `userPrincipalName` might be different from the domain name that uniquely identifies the user. Similarly, a user's logon name (`sAMAccountName`) might be different from the user name used in the `userPrincipalName`. For example, if the Active Directory user `sofia.perez@sierra.org` has a user logon name of `SIERRA\perez.s`, that user would be found as `perez.s@sierra.org`.

Adding multiple profiles for a user to a zone

It is possible for a single Active Directory user to have more than one UNIX profile defined in a zone. If you attempt to add a new UNIX profile for an Active Directory account that already has a UNIX profile in the current zone, Access Manager displays a warning but allows you to continue.

If an Active Directory user has more than one UNIX profile in a zone, however, the user should log on to computers in the zone with the UNIX profile name he wants to use. Logging on with the Active Directory user login name—the user's `sAMAccountName` attribute—might prevent the user from accessing some files because the account has multiple UNIX profiles and UIDs associated with it. In most cases, users can log on with their Active Directory account name if you have created parent and child hierarchical zones that address conflicting profile attributes. However, if you are using classic zones or hierarchical zones that don't address the need for multiple UNIX profiles, users might encounter file ownership issues.

Enabling and disabling users in classic zones

If you have added user profiles to classic zones, you can enable or disable their UNIX profiles in those zones at any time. Enabling and disabling a UNIX profile is not applicable in hierarchical zones.

To enable or disable the UNIX profile for multiple users in a classic zone, select all of the user names to enable or disable using the `CTRL` or `SHIFT` keys, right-click, then click **Enable UNIX Account** or **Disable UNIX Account**.

Forcing replication for read-only domain controllers

If the Active Directory forest includes read-only domain controllers, you should force replications when adding or modifying users and groups in a zone.

Forcing replication ensures that the new information is available right away.

To force replication after updating a zone:

1. Click Start > Administrative Tools > Active Directory Sites and Services.
2. Expand Sites, then select the Active Directory site that contains the connection over which you want to replicate directory information.

   For example, select **Default-First-Site-Name**.
3. Expand **Servers**, then select the read-only domain controller for which you want to force replication.
4. Click **NTDS Settings**.
5. In the details pane, right-click the connection over which you want to replicate directory information, then click **Replicate Now**.

If you choose not to force replication, the changes made to the zone will not take effect until replication is complete for the forest.

Using configuration parameters and group policies

You can use local configuration parameters or applied group policies to manage many operations for users and groups on Linux and UNIX computers. For example, you can use configuration parameters or group policies to bypass Active Directory authentication for specific users or to allow some users or groups to be approved for prevalidation. For more information about working with group policies, see the *Group Policy Guide*. For more information about setting parameters in the Centrify configuration file, see the *Configuration and Tuning Reference Guide*.

Enabling and configuring local account management

The local account management features described earlier in this guide require that local account management be enabled and configured.

Several configuration parameters and group policies let you control whether local account management is enabled in your environment, and how local account management is configured after it is enabled.

Local account management is disabled by default unless you are upgrading from a release in which local account management was enabled.

Follow these guidelines to determine whether you need to enable local account management:

- If you perform a fresh installation of Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service, the **Enable Local Account Management Feature** group policy is set to **Disabled**, and the `adclient.local.account.manage` configuration parameter on each local (agent-managed) computer is set to `false`. To use the local account management features described in this guide, you must manually enable local account management by setting the **Enable Local Account Management Feature** group policy to **Enabled**, or by setting the `adclient.local.account.manage` configuration parameter to `true`.

  See the following sections, "Group Policies" and "Configuration Parameters," for more information.

- If you are upgrading from a previous release, you can check the **Enable Local Account Management Feature** group policy setting to enable or disable local account management.

The following information is a summary of how various parameters and group policies affect local account management enablement and configuration. For more details about these parameters and group policies, see the *Configuration and Tuning Reference Guide* and the *Group Policy Guide*.

Group Policies

- **Enable Local Account Management Feature**: Use this group policy to control whether local accounts are managed by the UNIX agent and Access Manager. This group policy is disabled by default, unless you are upgrading from a previous release in which local account management was enabled.

  This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Local Account Management.

  This group policy controls the `adclient.local.account.manage` configuration parameter.

- **Notification Command Line:** Use this group policy to define a command to process changes to local account profiles after the agent synchronizes local user and group profiles with profiles defined in Access Manager.

This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Local Account Management.

This group policy controls the `adclient.local.account.notification.cli` configuration parameter.

- **Set refresh interval for access control cache**: Use this group policy to specify how often `etc/group` and `etc/passwd` are updated on UNIX and Linux computers, based on the local group and local user settings that you configure in Access Manager. This group policy also controls how often the authorization store cache is updated.

  This group policy is located in Computer Configuration > Centrify Settings > DirectControl Settings > Network and Cache Settings.

  This group policy controls the `adclient.refresh.interval.dz` configuration parameter.

Configuration Parameters

The following configuration parameters are located in the `/etc/centrifydc/centrifydc.conf` configuration file.

- **adclient.local.account.manage**: Use this parameter to control whether local account management is enabled on an individual computer. This parameter has a value of `false` by default, unless you are upgrading from a previous release in which local account management was enabled.
- **adclient.local.account.notification.cli**: Use this parameter to define a command to process changes to local account profiles after the agent synchronizes local user and group profiles with profiles defined in Access Manager.
- **adclient.local.account.notification.cli.arg.length.max**: Use this parameter to specify the maximum argument length for the command that you define in the `adclient.local.account.notification.cli` parameter.
- **adclient.refresh.interval.dz**: Use this parameter to specify how often `etc/group` and `etc/passwd` are updated on an individual computer based on the local group and local user settings that you configure in Access Manager. This parameter also controls how often the authorization store cache is updated.

Authorizing basic access

This chapter describes the basic principles of authorization and how to grant access to Centrify-managed computers using the default predefined rights and role definitions for Linux and UNIX computers. You should review the information in this chapter before creating custom role-based access rights and role definitions.

Basic concepts of access rights and roles

To log on and use Centrify-managed computers, Active Directory users must have a complete UNIX profile and be assigned to at least one role that grants them access. Both the profile and the role assignment can be explicitly defined for the zone or for an individual computer, or inherited from a parent zone.

You can use Access Manager to centrally manage what users can do on computers that have the Centrify agent installed. For example, you can control who can log on or connect remotely for each computer in a zone through the definition of rights and the assignment of roles. A **right** represents a specific operation that a user is allowed to perform. A **role** is a collection of rights that can be defined in a parent or child zone and assigned to Active Directory users and groups.

The most basic rights are the predefined **system rights** that determine whether a user can log on locally with a password, log on remotely without a password, and run commands in a standard shell or in a restricted shell. The most common settings for these system rights are defined by default in the UNIX Login role so that you can grant users access to Centrify-managed computers by simply assigning the predefined UNIX Login role and without defining any custom roles or creating any additional access rights.

System rights authorize access in role definitions

System rights are always associated with a role definition, whether it is a predefined role such as the UNIX Login role or a custom role you create. You can enable or disable specific system rights in any role definition, but you cannot add, modify, or delete the rights themselves. For Linux and UNIX computers, you can select the following system rights for any role:

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

In addition to the platform-specific system rights, there is a system right that allows users to bypass auditing or role restrictions to log on when there are problems on a computer. By selecting the **Rescue rights** option you can allow users in a particular role to log on in situations when all users would normally be locked out. For example, if authentication, authorization information, or auditing is required but not available, most users are prevented from logging on. You can use the rescue rights option to allow selected administrators to access the computer and fix the issues that are preventing other users from logging on.

Note: If you do not explicitly set the **Allow users assigned to this role to log on if problems with authentication, authorization or auditing services prevent logon access** rescue right option for any users, only the local `root` account will have rescue rights. The `root` account is always allowed to log on by default.

Access rights defined in the UNIX Login role

The predefined UNIX Login role is configured by default to allow users to log on locally with a password, connect remotely to a computer without being prompted for a password, and access the standard shell environment. The UNIX Login role is also configured to allow users to access all PAM-enabled applications in their environment. The UNIX Login role grants access to PAM-enabled applications through a predefined `login-all` PAM access right.

For most users and organizations, the default settings in the UNIX Login role make the user experience consistent before and after deploying the Centrify agent and joining an Active Directory domain. Users can log on and use the shell environment and applications in the same way they did before the deployment of the Centrify agent.

The predefined UNIX Login role and predefined `login-all` PAM access right are available by default in every zone. Depending on your requirements and policies, you can assign the UNIX Login role to all Active Directory users or to specific Active Directory users and groups. You can also choose whether to assign the UNIX Login role in parent or child zones to control where different groups of users can log on to Linux and UNIX computers.

Users must have both a complete identity profile and at least one role assignment that grants access before they can log on to any Centrify-managed computer. If you don’t use the UNIX Login role, you must create at least one custom role definition that provides similar functionality.

Default access rights and roles

In addition to the predefined UNIX Login role that grants basic access to Centrify-managed computers during deployment, there are other predefined access rights and role definitions that are available by default in every zone. These other predefined rights and role definitions provide specialized access rights for specific scenarios that are common in Linux and UNIX environments.

Default PAM access rights

For Linux and UNIX computers, the following predefined PAM access rights are available:

- `login-all` grants access to all PAM-enabled applications by specifying the asterisk (*) wild card for the application name. This right is included in the predefined UNIX Login role. You can add this right to any custom role to grant access to all PAM applications, such as `login`, `ftp`, `ssh`, `telnet`, and many others, without specifying them individually.
- `ssh` grants access to secure shell sessions on Debian and Ubuntu 6 and 7 computers. By default, this access right grants users access to all secure shell applications and operations.
- `sshd` grants access to secure shell sessions on all Linux and UNIX computers except Debian and Ubuntu 6 and 7 computers. By default, this access right grants users access to all secure shell applications and operations.

Default secure shell (SSH) access rights

Secure shell (SSH) access rights enable you to limit what users who are granted the PAM `ssh` or `sshd` right can do. These rights have no effect without the PAM `ssh` or `sshd` right. In addition, the default secure shell rights are only applicable for the Centrify-compiled version of OpenSSH.

For Linux and UNIX computers, the following predefined secure shell access rights are available:

- `dzssh-all` grants access to all secure shell services.
- `dzssh-direct-tcpip` allows local and dynamic port forwarding (`ssh-L`, `ssh -D`).
- `dzssh-exec` allows command execution.
- `dzssh-scp` allows secure copy (scp) operations.
- `dzssh-sftp` allows secure file transfer (sftp) operations.
- `dzssh-shell` allows secure terminal (tty/pty) connections.
- `dzssh-Subsystem` allows an external subsystem except `sftp` subsystem which has its own right.
- `dzssh-tcpip-forward` allows remote port forwarding (`ssh -R`).
- `dzssh-tunnel` allows tunnel device forwarding.
- `dzssh-X11-forwarding` allows X11 forwarding.

Predefined role definitions

In addition to the predefined UNIX Login role, there are several predefined role definitions that are available by default in every zone. For Linux and UNIX computers, the following predefined role definitions are available:

- `listed` makes a user profile visible in a zone but does not grant any type of access rights, PAM rights, or command rights. This is a specialized role that can be used when a user profile must exist for computers in a zone, but no local or remote access should be granted. For example, if a user owning files on a computer in a zone should no longer have access to the computers in the zone, you can assign the listed role so that the files continue to have an owner, but the user has no effective logon rights in the zone.
- `local listed` makes a local user profile visible in a zone but does not grant any type of access rights. This is a specialized role that can be used when a user profile must exist for computers in a zone, but no user access should be granted. For example, if a user owning files on a computer in a zone should no longer have access to the computers in the zone, you can assign the listed role so that the files continue to have an owner, but the user still has no effective rights in the zone.
- `require MFA for login` forces two-step authentication for access. This role does not grant access to any PAM applications but can be used in combination with the `UNIX Login` role to require users who are assigned to both roles to provide more than one form of authentication. You can also use this role with custom roles that grant access to specific applications if you want to require multi-factor authentication for those applications. You should note that using this predefined role definition requires additional configuration outside of Access Manager. For more information about what is required to support multi-factor authentication, see Requiring multi-factor authentication to log on.
- `Rescue - always permit login` enables users to log on to computers if there are problems with the authentication, authorization, or auditing service that are preventing other users from logging on. For example, if auditing is required on a computer and the auditing service is not available, only users assigned to a role with the “rescue” system right will be able to log on.
- `scp` grants secure copy (scp) access rights.
- `sftp` grants secure file transfer (sftp) access rights.

Identifying the scope for role definitions

The rights from multiple role assignments accumulate, which provides great flexibility and granularity in how you define and assign rights and roles. For example, you can use the UNIX Login role to control basic access, and define a second role that grants the rights to execute a set of privileged commands, so that a user assigned to both roles could log on, but only execute a few specific commands with elevated privileged. By separating rights into separate role definitions, not every role requires PAM applications or system rights, as long as a user is assigned a role that has those rights.

Because access rights are additive, however, it is important to consider where you define and assign roles to control who has administrative privileges on which computers. For example, it might seem reasonable to assign the predefined UNIX Login role to all Active Directory users. Doing so, however, could grant broad permission to log on to Linux or UNIX computers to which you want to restrict access. If you assign that role in a parent zone, it is inherited along with any additional rights granted in child zones.

In most cases, it is appropriate to define roles in parent zones, but assign roles carefully in child zones to avoid granting access rights on computers that host administrative applications or sensitive information.

Assigning the UNIX Login role

The predefined UNIX Login role allows Active Directory users to log on to Centrify-managed computers using any PAM-enabled application—such as `login`, `ssh`, or `ftp`—with a default shell and permission to execute the same set of commands available to any standard UNIX user account. By default, the UNIX Login role is configured to take effect immediately and never expire. By default, the UNIX Login role is also configured to audit user activity if the auditing service is running on a computer users access.

The default settings are appropriate for most Linux and UNIX users in most organizations. However, you can change any of the default settings in either a parent or a child zone, if needed.

What to do before assigning the UNIX Login role

You can assign the UNIX Login role to all Active Directory users, to specific Active Directory users, or to specific Active Directory groups. Because the UNIX Login role is a predefined role, you cannot assign any local users to the role.

Before you assign the role, you should decide whether you want to assign and inherit the role from a parent zone or make the assignment in a specific child zone. You should also decide whether you want to specify optional start and end times for some role assignments.

Rights required for this task

The following table describes the minimum rights that must be granted for users to successfully manage role assignments in a zone:

This target object	Requires these permissions	Applied to
`Authorization`	On the **Object** tab, select **Allow** for the following: ■ List contents ■ Read all properties ■ Create all child objects ■ Delete all child objects ■ On the **Properties** tab, select **Allow** for the following: ■ Write msDS-AzApplicationData	This object only
	On the **Properties** tab, select **Allow** for the following: ■ Write displayName ■ Write msDS-AzApplicationData ■ Write msDS-TasksForAzRole ■ Write msDS-MembersForAzRole	The msDS-AzRole object

This target object	Requires these permissions	Applied to
`AzRoleObjectContainer`	On the **Object** tab, select **Allow** for the following: ■ List contents ■ Read all properties ■ Create msDS-AzRole objects ■ Delete msDS-AzRole objects	The msDS-AzApplication object and all child objects
	On the **Properties** tab, select **Allow** for the following: ■ Write displayName ■ Write msDS-AzApplicationData ■ Write msDS-TasksForAzRole ■ Write msDS-MembersForAzRole	The msDS-AzRole object
	On the **Properties** tab, select **Allow** for the following: ■ Write msDS-AzApplicationData	The msDS-AzAdminManager object

This target object	Requires these permissions	Applied to
`AzOpObjectContainer`	On the **Object** tab, select **Allow** for the following: ▪ Read all properties ▪ Create msDS-AzOperation objects ▪ Delete msDS-AzOperation objects ▪ Create msDS-AzRole objects ▪ Delete msDS-AzRole objects	This object only
	On the **Properties** tab, select **Allow** for the following properties: ▪ Write displayName ▪ Write msDS-AzApplicationData ▪ Write msDS-TasksForAzRole ▪ Write msDS-MembersForAzRole	The msDS-AzRole object
	On the **Properties** tab, select **Allow** for the following: ▪ Read name ▪ Read Name ▪ Write msDS-AzApplicationData ▪ Write name ▪ Write description	The msDS-AzOperation object

Who should perform this task

A UNIX administrator who manages one or more zones most often performs this task, depending on your organization’s policies.

• • • • • •

How often you should perform this task

In most organizations, you assign the UNIX Login role to target groups of users at a time during deployment and as needed, thereafter.

Steps for completing this task

The following instructions illustrate how to assign the UNIX Login role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To assign users and groups to the UNIX Login role in a zone

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to make role assignments.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the UNIX Login role definition from the list of roles, then click **OK**.

   By default, the role is set to start immediately and never expire. You can set a **Start time**, **End time**, or both start and end times for the role assignment. For example, if the role assignment applies to a contractor who will be hired for a specific period of time and you want to automatically disable the role after they finish the job and leave the organization, you can specify the start and end times when you assign the role.
6. Select whether the role assignment applies to all Active Directory accounts or specific accounts.

   If you want to automatically assign the role to every user added to the Active Directory forest or trusted forests, you can select **All Active Directory accounts** for convenience. This option is similar to selecting the "Authenticated Users" or "Everyone" system groups. For example, if you want to assign all Active Directory users the UNIX Login role by default, you can select this option. Only users who also have a complete UNIX profile will be able to log on to the UNIX computers joined to the domain.

If you are assigning the role to specific accounts, click **Add AD Account** to search for and select the Active Directory groups or users to assign to the role, then click **OK**.

7. Click **OK** to complete the role assignment.

What to do next

Verify Active Directory users or group members assigned the UNIX Login role can log on to Centrify-managed computers in the zone where you have made the role assignment.

Where you can find additional information

If you want to learn more about working with rights, roles, and role assignments, see the following topics for additional information:

- Defining rights to use commands
- Defining rights to use PAM applications
- Using secure shell session-based rights
- Creating and assigning custom role definitions

Performing role assignment on multiple computers

To simplify the process of assigning Active Directory users or groups to a role, you can perform a bulk role assignment. With a bulk role assignment, you can assign a role to multiple Active Directory users and groups on multiple computers at the same time. For example, if you have two groups of Oracle administrators and three computers where the members of those groups need access to their OracleAdmin role, you can select those two groups and those three computers to be assigned the OracleAdmin role in the same process. You can also specify optional start and end times for the role assignment and have those settings apply for all of the users, groups, and computers you have selected for bulk assignment.

To assign a role to multiple users and groups on multiple computers

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to make a bulk role assignment.
3. Right-click, then select **Assign Roles to Computers**.
4. Type the user and group names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for user and group names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria. If multiple users or groups match the search criteria, select the appropriate users and groups, then click **OK**.
5. Type the computer names you want to be included in the role assignment, then click **OK**.

   You can specify multiple names separated by a semi-colon (;). You can also search for the computer names by typing part of the name and clicking Check Names or by clicking Advanced and entering search criteria. If multiple computers match the search criteria, select the appropriate computers, then click **OK**.
6. Select a role for the list of roles available, then click **OK**.
7. Review the role assignment start and end time and the user and group accounts that are being assigned the role, then click **OK**.

   You can make changes to the start and end times if you want those changes applied for all of the users, groups, and computers that are part of this bulk role assignment.

After you click OK, the selected users and groups are then automatically assigned the selected role on the selected computers.

Viewing rights and roles

Access Manager allows you to view the status and effective rights for any Active Directory user or local user in a zone, whether they have been assigned a role or not. You can view detailed information about the rights and role assignments for users by using **Show Effective UNIX User Rights**. If a user is not assigned a role or does not have a complete user profile, be certain to select the **Show omitted users** option, otherwise, information will not be shown for the user.

> **Note:** Local users are defined in Access Manager in the zone and are saved in `/etc/passwd` on each computer in each zone where the profile is defined. Local users that you define in the zone do not need to be Active Directory users. For more information about local users, including information that is required for a user profile to be complete, see Creating user profiles.

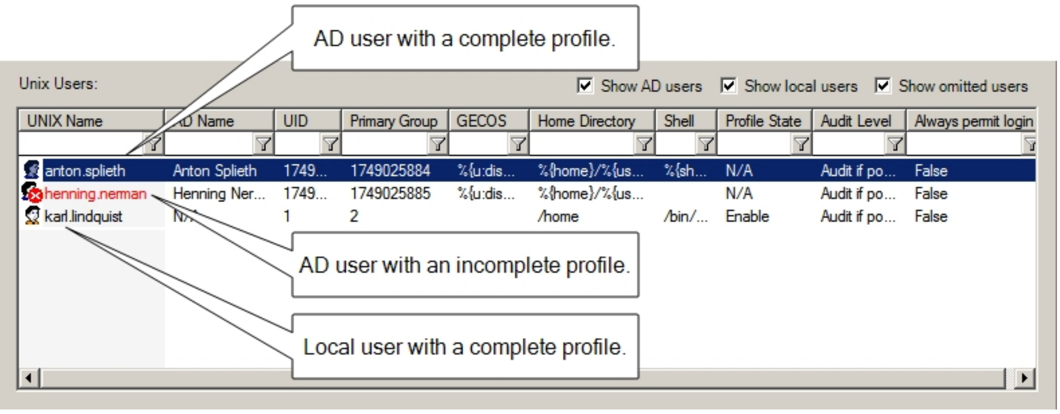
To view rights for an individual user in Access Manager

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to view rights and other account details.
3. Right-click, then select **Show Effective UNIX User Rights**.
4. Select a computer or click Browse if you want to limit the information included to a specific computer.
5. Select **Show AD users** and **Show local users** as necessary, depending on which users you want to view. One or both of these choices might already be selected, depending on the location from which you originally selected **Show Effective UNIX User** rights.
6. Select **Show omitted users** to include users who have an incomplete profile or do not have a role assignment in the list of UNIX users.

   User information is displayed as shown in the following example. Key points about the information displayed are as follows:

   - Users with incomplete profiles are displayed in red (if **Show omitted users** is selected).
   - Local users are not required to have an AD name, resulting in a displayed **AD Name** value of `N/A`.
   - AD users are not required to have a UNIX profile, resulting in a displayed **Profile State** value of `N/A`.
   - For more information about the differences between AD users and local users, as well as details about profile states for local users, see

Creating, modifying, and deleting user profiles for local users.



AD user with a complete profile.

AD user with an incomplete profile.

Local user with a complete profile.

Unix Users: ☑ Show AD users ☑ Show local users ☑ Show omitted users

UNIX Name	AD Name	UID	Primary Group	GECOS	Home Directory	Shell	Profile State	Audit Level	Always permit login
anton.splieth	Anton Splieth	1749...	1749025884	%{u:dis...	%{home}/%{us...	%{sh...	N/A	Audit if po...	False
henning.nerman	Henning Ner...	1749...	1749025885	%{u:dis...	%{home}/%{us...		N/A	Audit if po...	False
karl.lindquist	N/A	1	2		/home	/bin/...	Enable	Audit if po...	False

7. Select a user to see more detailed information about the user's profile, role assignments, and rights in the selected zone or on a specific computer:

   - Click **Zone Profile** to review the UNIX profile defined for a user and where the profile attributes are defined. If a user has an incomplete profile, you can click the **Zone Profile** tab to see which profile attributes are missing.

   - Click **Role Assignments** to review a user's role assignments. The Object Assigned column indicates whether the role is explicitly assigned to the user (user@domain) or to a group the user is a member of (group@domain). The Location of Assignment column indicates the zone or computer role in which the assignment was made. Information for the Start Time, End Time, or both columns is only displayed if a role assignment has time constraints.

   - Click **PAM Accesses** to review the PAM application access rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.

   - Click **Commands** to review the command access rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.

   - Click **SSH Rights** to review the secure shell rights for the user in the selected zone or on a specific computer, including the role to which the right belongs.

8. Click **Close** when you are finished reviewing user rights in a zone or on particular computers.

• • • • • •

Checking rights and roles with the dzinfo program

You can also view rights and roles for specific users or the current user by running the `dzinfo` command-line program on Centrify-managed computers. If you want to use the `dzinfo` program to view roles and rights for other users, however, you must have `root` permission.

You can run `dzinfo` without any arguments to see your own rights and role assignments. The command displays detailed information about the your role assignments, the availability for each role assignments, your effective rights, the current audit level, and the specific PAM access, command, and secure shell rights you have been granted.

To see more detailed information, such as the days and times a role is available, you can use the `--verbose` option. For example, to see detailed information, you could type the following command:

```
dzinfo --verbose
```

To view roles and rights for a specific user:

1. Log on or switch to `root` on a managed computer.
2. Run the `dzinfo` command for a specific user with the username in the command line.

   ```
   dzinfo username
   ```

   For example, to see details about the rights and roles assigned to the user `sonya`, you could type the following command:

   ```
   dzinfo sonya
   ```

If rights and role assignments have been configured for the specified user, the command displays detailed information about the user's role assignments, the availability of those role assignments, the user's effective rights, the audit level in effect, and the specific rights that have been granted.

You can also use the `dzinfo` program to test whether a user has the right to run specific commands. For more information about using `dzinfo` and the `dzinfo` command line options, see the `dzinfo` man page.

Changing the audit level for role definitions

By default, all role definitions—including predefined role definitions—are set to "Audit if possible" as the audit level. With this setting, user activity is audited if the auditing service is installed and enabled on a managed computer. If the auditing service is not installed or not running on a given computer, this setting has no effect. Users can log on and use the access rights that are defined for their role assignment without having their activity audited.

In most cases, the default "Audit if possible" setting is appropriate because it doesn't block user access if you are not deploying the auditing infrastructure but will automatically capture user activity if you are deploying auditing. In some cases, however, you might want to change the audit level. You can modify the audit level for any role definition to specify whether users must be audited in order to log on.

To change the audit level for a role definition:

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to change the audit level.
3. Expand Authorization and Role Definitions.
4. Select a role definition, right-click, then select **Properties**.
5. Click the **Audit** tab.
6. Select the appropriate audit level to use for the role definition.
    - Select **Audit not requested/required** if you are not interested in auditing session activity for users in the role.
    - Select **Audit if possible** if you want to audit user activity on computers running the Centrify auditing service. If you select this option and the auditing service is not installed or not currently available, users assigned to the role are allowed to log on without having their activity audited. This option is selected by default for new roles.
    - Select **Audit required** if you want to audit all session activity for users assigned to the role. If you select this option and the auditing

service is not installed or not currently available, users assigned to this role are not allowed to log on.

If auditing is required for users in a role, you should also define a role with rescue rights to allow selected administrators to log on and correct problems when other users are locked out. For more information about creating a role with rescue rights, see Creating a role definition with rescue rights.

7. Click **OK** to save the role definition.

Requiring multi-factor authentication to log on

You can configure multi-factor authentication for users logging on to Centrify-managed computers to improve the security of physical or virtual data centers. You can assign the predefined `require MFA for login` role in combination with the `UNIX Login` role to require users who are assigned to both roles to provide more than one form of authentication. You can also create custom role definitions with the **Require multi-factor authentication for login** system right. Before setting this system right, however, you should be aware the multi-factor authentication for Centrify-managed computers relies on the infrastructure provided by the Centrify identity platform and Centrify identity services.

As a preview, here are the steps involved to enable multi-factor authentication for Centrify-managed computers in hierarchical zones:

- Register for Privileged Access Service.
- Install and configure at least one **connector** for communication with Privileged Access Service.
- Verify the users who are required to provide more than one form of authentication have valid **Active Directory accounts** that are active in Privileged Access Service.
- Add or select the **authentication profiles** that specify the types of authentication challenges to support.
- Create a role with the appropriate **computer members and administrative rights** for multi-factor authentication.
- Verify the **identity platform instance URL** you want to use if you have access to more than one instance.

After you have completed the preliminary steps, you can assign users the predefined `require MFA for login` role or a custom role with the **Require**

multi-factor authentication for login system right to require two-step authentication when logging on using PAM applications. These preliminary steps are also required if you want to create command rights that require two-step authentication when executing commands using elevated privileges (`dzdo`) or in restricted shell (`dzsh`) environments.

The preliminary steps are also required to support multi-factor authentication in classic zone and Auto Zone. However, the implementation is slightly different than in hierarchical zones, so some of the steps differ depending on the type of zone where you want to use multi-factor authentication. For more information about preparing to use multi-factor authentication, see "Preparing to use multi-factor authentication" in theMulti-factor Authentication Quick Start Guide.

Defining rights to use commands

As discussed in Basic concepts of access rights and roles, access rights allow users to perform specific operations. You define the most basic rights—such as the right to log on or connect remotely—when you define roles. However, you can use more granular command access rights to tightly control who has access to individual command-line programs. This chapter describes how to define access rights that allow users to execute command-line programs on Centrify-managed computers.

Controlling access to commands

In a standard UNIX shell environment, an ordinary user account can execute a large number of common command-line programs without any special privileges, and one or more administrative accounts, such as `root`, are required to execute commands that perform privileged operations. If ordinary users need to execute any of the commands requiring administrative privileges, they might have to switch to an administrative account that requires them to know the password for a privileged users or been granted access by configuration settings in a `sudoers` file.

For Centrify-managed Linux and UNIX computers, however, you can define command access rights to tightly control the specific commands users can execute. You can also refine those rights to only allow specific arguments to be used or to require an executable to be located in a specific directory.

There are no predefined rights for commands. Therefore, only the specific command access rights you define will be available for you to add to roles. You should keep in mind that any command rights you define are specific to the zone where you configure them, but can be used in any child zones of that zone.

What command rights provide

Command access rights identify the specific commands that can be executed on a Linux or UNIX computer by a user assigned the role to which the rights are added. Command rights also specify whether the commands defined in the right are executed under the user's own account or using another user account.

There are two primary reasons for defining command rights:

- To **grant access** to specific commands that must be executed with elevated privileges
- To **restrict access** to only allow specific commands to be executed.

Granting access using command rights

The most common reason for defining a command right is to grant access to commands that perform privileged operations. For example, you might want to grant users additional privileges to execute specific commands in a standard shell environment that they are not otherwise allowed to execute with the default rights associated with their account.

With this type of command right, most commands are executed in the default shell environment with ordinary user privileges. When users assigned to a role with this type of command right want to use their elevated privileges, they invoke the command they have been granted access to using the `dzdo` command. This type of command right is similar to configuring privileges in a `sudoers` file, then invoking a command using `sudo`.

This type of command right is appropriate for UNIX users who have a standard shell environment and only need elevated rights to perform specific tasks.

Restricting access using command rights

It is less common, but also possible to define a command right to restrict access. For example, you might want to create a role that provides strictly controlled access to an explicitly defined subset of shell commands. This type of command right creates a customized restricted environment shell (`dzsh`) where only explicitly defined commands can be executed. This type of command right is similar to configuring a "whitelist" of allowed command and is appropriate for users who only need access to a limited set of commands to perform their job.

Controlling the shell environment for commands

You can define command rights to control who has permission to run specific commands in a zone. When you define individual command rights, you can also specify whether the commands can be executed in a non-restricted shell environment, a restricted shell environment, or both. After you define the command right, you can then add it to an appropriate role definition. It is then the role definition to which you add the command right that controls whether users can use the command in a standard, unrestricted shell environment or in a restricted shell environment.

If the role definition allows a non-restricted shell environment—like the UNIX Login role—the command right provides functionality similar to the UNIX `sudo` command except that it uses the role settings and the zone authorization store rather than through a `sudoers` configuration file.

If the role definition does not allow access to a non-restricted shell environment, the command right can only be used in a restricted shell environment and users assigned to the role can only execute the specific commands explicitly defined in command right.

Defining rights to run privileged commands

The most common reason for creating a command right is to allow users to execute commands that require privileges not granted to a standard UNIX user account. For example, you might want to grant some users permission to run Centrify command-line programs that require `root` privileges to better manage their own computers.

Defining command rights that grant elevated privileges is similar to granting access to privileged commands using the `sudoers` configuration file and the `sudo` program.

Steps for completing this task

The following instructions illustrate how to define a command right to execute a command with elevated privileges. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer’s Kit*, or in community forums on the Centrify website.

To define a command right for privileged access

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to define a command right.
3. Expand Authorization and UNIX Right Definitions, then select Commands.
4. Right-click, then click **New Command**.
5. On the General tab, type a short descriptive name for the command right, and optionally, a more detailed description for the command right.

   The privileged command name is required and must not be more than 63 characters in length or contain any special characters, such as asterisks (*), slashes (\ /), question marks (?), or quotation marks (“).
6. Type the command you want to add.

   The Command field is required and should include any parameters or options, if needed. You can also use wild cards or a regular expression to specify commands matching a particular pattern.
7. Select the type of pattern matching to use for the “Command” and “Specific path” fields.

   - Select **Glob expressions** to use `glob` pattern matching syntax for wild cards.
   - Select **Regular expressions** to use extended regular expression pattern matching.

   For more information about pattern matching, see Selecting the pattern matching syntax.
8. Select an appropriate path for matching the command on the different operating environments you support.

- Select **Standard user path** to use the local operating system's common set of user directories to find the command.
- Select **Standard system path** to use the directories the `root` user would normally get on the local operating environment to find the command.
- Select **System search path** if you want to search for the command in a predefined set of locations. The search locations are defined using the `dzdo.search_path` configuration parameter. If you select **System search path** and the `dzdo.search_path` parameter is not defined, the current user's path is used to search for the command.
- Select **Specific path** if you want to define a custom set of locations for finding the command specified. If you select this option, you can specify one or more paths, separated by a colon.

If you are specifying a path, the path must start with a forward slash (/) unless you are matching all paths (*). For example, if the command you specify is `ls` and you set the path to *, the `ls` command from any path is allowed.

If you set both the "Command" field and the "Specific path" field to match all strings (*), any command from any path is allowed.

9. Specify an integer that determines the priority of the command — the lower the number, the higher the priority.

   If there are multiple commands that match the pattern you specified for the "Command" field, the priority determines which command has higher priority.

10. Click the **Run As** tab, then select **Can be used by dzdo** to allow the command to be added to a role for privileged execution.
11. Select the user or group accounts that can be used to execute the command.

    - Select **Any User** if any standard user account can be used to execute the command with dzdo.
    - Select **One of the following users, uids, groups or gids** if you want to specify one or more user or groups that can be used to execute the command with dzdo.

    In most cases, the local `root` account is the appropriate account to use because it allows ordinary users to execute the specified command using `root` account privileges. However, you can click **Add** to add other users, groups, or service accounts that can be used to execute the command. Use the format #UID for UID values, %group for group names, or %#GID for GID values.

The account used to execute commands can be an Active Directory user with a UNIX profile in the zone or a local UNIX user account. However, the account used to log on and invokes the command using `dzdo` must be associated with an Active Directory account.

Optionally, you can specify the primary groups can be used when executing the command using `dzdo`:

- Select **Any Group** if any group can be used as the primary group when executing the command with `dzdo`.
- Select **One of the following groups**, then click **Add** if you want to specify the groups that can be used as the primary group when executing this command with `dzdo`.

You can also configure commands to be executed using `dzdo` in a restricted shell environment. For this example, however, the command right is only used in a non-restricted shell environment.

12. Click **OK** to save the new command right.

    In most cases, you can use the default settings for environment variables and execution attributes.

    - If you want to keep, remove, or add environment variables for command execution, see Customizing environment variables for command execution.
    - If you want customize any of the execution attributes, see Customizing command execution attributes.

Creating a role to run commands with elevated privileges

On most Linux and UNIX computers, you can identify commands that require elevated permissions, who can run those commands, and where different users or groups can run the commands using a `sudoers` configuration file. Users who have been granted the appropriate permissions can run privileged commands by invoking the `sudo` command.

Centrify provides similar functionality, but the commands are configured by defining command rights, adding the rights to the appropriate roles, and assigning the roles to different users and groups. Users who have been assigned the appropriate roles can then run privileged commands by invoking the `dzdo` command.

If users are assigned the predefined UNIX Login role, they have access to all of the standard command-line programs that are available to ordinary UNIX users. You can create a separate role for commands that run using `root` or another privileged user account. Alternatively, you can combine command rights and system rights in a custom role definition or by adding the command rights to the default UNIX Login role.

Command rights that allow users to execute commands with elevated privileges should only be added to roles with the **Login with Non-Restricted Shell** system right.

Users must execute command rights that grant elevated privileges using the `dzdo` command. If you selected the **Re-authenticate current user** option as an execution attribute when defining a command right, users must also provide the password for their own account, their own password and one or more other forms of authentication, or the types of authentication determined by the authentication profile configured in Privileged Access Service, which might or might not involve providing a password.

If you selected the **Re-authenticate using the target user's password** option as an execution attribute when defining a command right, users must also provide the password for the account used to execute the command.

To create a role that can execute commands with elevated privileges, do the following:

- Create command rights for the privileged commands users are allowed to run.
- Create a new role definition and set the System Rights for the role to allow password login, non-password login, or both, and select the **Login with Non-Restricted Shell** option, then click **OK** to save the role definition.
- Right-click the role, select **Add Right**, then select `login-all` or a specific PAM access right and the privileges command rights users are allowed to run, then click **OK** to save the changes to the role definition.

For more information about creating, assigning, and testing custom role definitions, see Customizing command execution attributes.

Defining a restricted shell command right

You can also use command rights to strictly control which commands certain users can execute. In a restricted shell environment, users can only execute the specific commands and command-line options that are explicitly allowed. For example, you might want to grant some users permission to run a specific Centrify command-line program, such as `adinfo`, without allowing them to run any other command-line programs on some computers.

Users who are assigned to a role with the restricted shell environment are not be able to run any other commands, including informational commands such `ls`, `ps`, and `whoami`, unless you explicitly include them in the command right. You are not required to explicitly add basic navigational commands, such as `cd` and `pwd`, to the command right.

What the restricted shell provides

For Linux and UNIX computers, Centrify provides a customized Bourne shell, `dzsh`, to serve as the restricted shell environment. The `dzsh` restricted shell supports environment variables, job control, command history, and the specific command rights you define. For example, you can use the up-arrow key in the `dzsh` shell to recall previously-entered commands. You can also set a limit to the command history available by adding `HISTSIZE=`n to the `$HOME/.dzshrc` file.

For most operations, working in the `dzsh` shell is similar to working in an unrestricted shell except that the command set available is limited to the command rights you add to the environment.

Limitations of the restricted shell

The restricted shell environment does not enforce rights for commands that run outside of the shell. For example, if users run a graphical desktop manager, they can run commands and applications that are launched from menu selections in the graphical user interface.

In addition, the command rights defined for the `dzsh` shell do not prevent users from running built-in shell commands, accessing the file system, or seeing process or system information. For example, even in a restricted shell environment with no rights to run any commands, users in a dzsh shell could get a process listing using the following script:

```
for i in /proc/[0-9]*;
  do read PROC < $i/cmdline;
  echo $PROC;
done
```

Because the shell scripting environment allows the operations, users can effectively access information that the commands defined for the restricted shell environment do not allow.

Securing the restricted shell environment

There are many ways sophisticated users can get around limitations placed on a restricted shell environment. For example, most text editors, such as vi and emacs, allow shell escapes. Giving users permission to run programs that allow shell escapes in a restricted shell enables them to open a new unrestricted shell environment with none of the restrictions placed on them in their defined environment, Similarly, giving users access to commands that set or modify local time and date settings might allow users to avoid time constraints for running commands or the expiration date and time for specific role assignments.

In some cases, even individual command line options might provide users with the means to run commands not defined in their restricted shell environment. For example, defining a command right that allows users to run the tar command with the --use-compress-program program_name option allows user to run the specified program_name even though the program_name is not an allowed command in their restricted shell environment.

In choosing the commands to allow in a restricted shell, therefore, you should carefully consider ways to plug potential security holes the commands might introduce or whether there are alternative commands that provide the same functionality more securely. For example, if you need to give a user access to an editor, such as vi or vim, you could restrict the ability to execute nested commands to prevent users from opening a new shell from within the editor. Alternatively, you could add the rvi command to the restricted environment instead of vi or vim because rvi doesn't allow the user to open a new shell.

For more information about setting attributes that control command executions, see Customizing command execution attributes.

Steps for completing this task

The following instructions illustrate how to define a command right for use in a restricted shell using Access Manager. For more information about any step, see Defining rights to run privileged commands. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a command right for restricted shell access

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name where you want to define a command right.
3. Expand Authorization and UNIX Right Definitions, then select Commands.
4. Right-click, then click **New Command**.
5. Type a short descriptive name for the command right, and optionally, a more detailed description for the command right.
6. Type the command you want to add.
7. Select the type of pattern matching to use for the "Command" and "Specific path" fields.
8. Select an appropriate path for matching the command on the different operating environments you support.
9. Specify an integer that determines the priority of the command—the lower the number, the higher the priority.
10. Click the **Restricted Shell** tab, then select **Can be used in a restricted role** to allow the command to be added to a role that runs in a restricted shell environment.
11. Select whether commands are executed using the user's logon account or using a specific the user name or UID.

    If you want to configure commands to be executed using dzdo in a restricted shell environment, you can click the Run As tab to specify a user or group for command execution.
12. Click **OK** to save the new command right.

    In most cases, you can use the default settings for environment variables and execution attributes.

- If you want to keep, remove, or add environment variables for command execution, see Customizing environment variables for command execution.
- If you want customize any of the execution attributes, see Customizing command execution attributes.

Creating a role to run commands in a restricted shell

For Linux and UNIX computers, Centrify provides a customized Bourne shell, `dzsh`, to serve as a restricted shell environment. The `dzsh` restricted shell supports environment variables, job control, command history, and the command access rights you define.

To create a role that runs a restricted shell, do the following:

- Create command rights for the restricted shell commands users are allowed to run.
- Create a new role definition and set the System Rights for the role to allow password login, non-password login, or both, and verify that the **Login with Non-Restricted Shell** option is not selected, then click **OK** to save the role definition.
- Right-click the role, select **Add Right**, then select `login-all` or a specific PAM access right and the restricted shell command rights users are allowed to run, then click **OK** to save the changes to the role definition.

For more information about creating, assigning, and testing custom role definitions, see Customizing command execution attributes.

Selecting the pattern matching syntax

When you define a command right, you can specify whether you want to use glob pattern matching syntax or extended regular expression syntax to match the strings specified for the “Command” and “Specific path” fields.

Using glob pattern matching

The default `glob` pattern matching enables you to specify a string using wild card characters. For example, with `glob` pattern matching, the command can contain a question mark (?) to represent any single character, an asterisk (*) to represent any string, including an empty string, or an expression enclosed by brackets ([. . .]).

You can also use an exclamation point (!) at the start of a command to disallow a matching string. For example, you can prevent users from specifying the program to use for viewing man pages (`man -P`) by specifying the following commands:

```
!man -P*
!man * -P*
man
```

The commands that start with the exclamation point take precedence over the ones that don't. For example, if you type "`!ls -l`" and "`ls *`" as command strings, users will be prevented from running the "`ls`" command with the "`-l`" option, even though "`ls *`" specifies that all options are allowed. If a command is followed by empty quotation marks (`""`), the command can only run without any options.

For more information about using `glob` for pattern matching, see the `glob` and `glob(7)` man pages.

Using regular expression pattern matching

If you select regular expressions for pattern matching, you must specify a valid regular expression for the "Command" and "Specific path" fields. Regular expressions consist of a combination of literal and special characters. For example, regular expressions can include backslash (\), caret (^), dollar sign ($), period (.), vertical bar (|), question mark (?), asterisk (*), plus sign (+), parentheses (( )), square brackets ([ ]), and curly braces ({ }) characters to define complex string matching patterns.

By default, regular expression commands are enclosed automatically with command "anchors" to prevent to prevent open-ended or unintended pattern matching for paths or commands. If you write precise regular expressions, you can disable the default behavior by setting the `dz.auto.anchors` parameter to `false` in the `centrifydc.conf` configuration file. If you set this parameter to `false`, however, you should carefully consider all of the possible matches for the regular expressions you define.

For more information about using regular expressions for pattern matching, see the `regcomp`, `regexec`, and `regex` man pages. For more information about setting configuration parameters, see the *Configuration and Tuning Reference Guide*.

Customizing environment variables for command execution

You can customize the environment variables used during command execution in both the non-restricted and restricted shell environments. For example, if a command is executed using a specific user or service account that requires environment variables that are not defined for the user invoking a command, you can define those environment variables as part of the command right definition.

If you want to configure the environment variables to use for a command right, click the **Environment** tab. You can then select one of the following options:

- Reset environment variables
- Remove unsafe environment variables
- Add environment variables

Resetting environment variables

Select **Reset environment variables** if you want to define the list of environment variables to set when the user runs the command. Note that only the environment variables you explicitly specify are retained and those environment variables will replace the default set of environment variables, rather than append the default set of environment variables. You can use Access Manager or `dzdo.env_*` configuration parameters in the `centrifydc.conf` file to control the list of environment variables to use when executing commands. For example, you can set the `dzdo.env_keep` configuration parameter in the `centrifydc.conf` file to keep a specific set of environment variables like this:

```
dzdo.env_keep: VAR
```

With this setting, only the `VAR` environment variable is defined for the list of environment variables to keep. All other environment variables, including the default list of user environment variables—such as `PATH` and `KRB5CCNAME`—are removed.

If you select this option, click **Edit** to specify the environment variables to retain from the user's environment in a comma-separated list. Click **Add**, type the environment variable name, then click **OK** for each environment variable you want to retain.

Removing environment variables

Select **Remove unsafe environment variables** if you want to remove a specific set of unsafe environment variables when the user runs the command. The list of unsafe environment variables is defined by the `dzdo.env_delete` configuration parameter in the `centrifydc.conf` file. Note that only the environment variables you explicitly specify are removed.

If you select this option, click **Edit** to specify the environment variables to remove from the user's environment in a comma-separated list. Click **Add**, type the environment variable name, then click **OK** for each environment variable you want to remove.

Adding environment variables

Select **Add environment variables** to define new environment variables to add when the user runs the command. Enter variables in a comma-separated list in the form name=`value`, or click **Edit** then **Add** to add new variables and values. You can add new variables regardless of which of the other options you select.

Customizing command execution attributes

You can modify the default command execution attributes that are used when commands run in either the non-restricted shell or in a restricted shell environment. In most cases, changes are rarely required for commands that run in a non-restricted shell. It is more common to change the execution attributes for commands that run in restricted shell environments. For example, you can use the execution attributes to control whether an allowed command can invoke a nested command. In a restricted shell environment, you might want to prevent a command from invoking nested commands to reduce the chance that users can run commands not explicitly defined for their environment.

If you want to set any execution attributes for a command right, click the **Attributes** tab. You can then select different options to control different aspects of command execution.

Requiring re-authentication to run commands

After successful authentication during the login process, you can control whether running a command in a restricted shell or using elevated privileges requires re-authentication or not. If you want to require re-authentication, select the authentication rules to apply. When defining the rights for executing commands, you can select from the following authentication options:

- No re-authentication required

  Select this option to allow users to run the command without any additional authentication.

- Re-authenticate current user

  Select this option to require the user to be re-authenticated before running the command using their own credentials. If you select this option, you can also specify whether re-authentication requires the user to provide their password, requires their password and another form of authentication, or requires multi-factor authentication as determined by the authentication profile configured in Privileged Access Service, which might or might not involve providing a password.

  If you select both **Use password** and **Require multi-factor authentication for login,** users are prompted to type their password and provide another form of authentication before the command is executed. If you have configured the authentication profile to accept more than one type of authentication challenge, users are prompted to select the authentication method to continue.

- Re-authenticate using the target user’s password.

  Select this option to require the user to be re-authenticated before running the command using the target run-as user’s credentials.

Preserving group membership

When defining command rights, you should consider whether keeping a user's existing group membership would provide benefits for command execution or could be exploited to perform unauthorized operations.Select **Preserve group membership** if you want to retain the logged-on user's group membership while executing commands.

Allowing nested commands

When defining command rights, you should consider whether allowing the execution of nested commands could be exploited to perform unauthorized operations. Select **Allow nested command execution** if you want to allow a command to invoke another program or open a new shell. To enhance the security of a restricted shell environment, you should deselect this option to prevent an allowed command to be used to run another program or open an unrestricted shell.

Preventing unsafe path navigation

When defining command rights, you should consider whether the command or any of the allowed command arguments could be exploited to perform unauthorized operations. One way command arguments can be exploited is to allow navigation up the path hierarchy. To prevent command arguments from allowing unsafe navigation up a path hierarchy, you can select the **Prevent navigation up a path hierarchy**. For example, if a command right allows a user to execute a command such as `vi /etc/httpd/conf/*` without this option, the right could be exploited by specifying a command argument that navigates up a path hierarchy to perform an unauthorized operation. In this case, the right might be used to edit any file as the `root` user by specifying a relative path as a command-line argument.

```
vi /etc/httpd/conf/../../shadowpass
```

You can avoid this potential security risk by disabling upward path navigation for command arguments, if needed. Note that this setting is only supported in hierarchical zones and is only applicable for glob command rights.

Setting the umask value

Set the **Umask value** by selecting the read (R), write (W), and execute (X) permissions for the owner, group, and other users if you want to change the permission settings for executing a command.

Setting SELinux role-based access control

Configure the **SELinux Setting** for dzdo Security Enhanced Linux (SELinux) role-based access control (RBAC). By enabling the SELinux role and SELinux type fields, privileged commands can be specified with the default role and type for creating SELinux context in execution. These settings can be overridden using the `'-r'`/`'-t'` command-line options respectively. To enable this setting, click the **SELinux Setting** button and enable SELinux role and SELinux type, then enter string values in the corresponding text fields. Settings are saved in the attribute of the `msDS-AzOperation` command object.

> **Note:** These settings are currently supported only on the RHEL systems and effective only on system with SELinux enabled and joined to a hierarchical zone.

Setting the command digest

You can use Digest Settings to specify SHA-2 digests so that sudo can verify the binary's checksum (SHA-2) before sudo executes the binary. The supported digest (hash) types are as follows:

- SHA224
- SHA256
- SHA384
- SHA512

Select a digest type, and then enter a checksum. You can specify multiple digests for a command.

Note that setting a command digest is only supported in the explicit path matches against the command right, and only supported in the hierarchical zone.

Testing command rights

After command rights have been defined, added to role definitions, and assigned, you can use `dzinfo --test “command”` to check whether you have permission to execute a specific command. You can use the `dzinfo` username `--test “command”` command to check whether a specified user has permission to run a specified command. If you want to use the `dzinfo` program to view command rights for other users, however, you must have `root` permission.

To check for command rights, you must enter the complete path to the command and enclose the command in single or double quotes. For example, to test whether the user, `qa1` has a command right that allows execution of the `id` command as `root`, you could run the following command on a Linux or UNIX computer:

```
[user1@rh5]# dzinfo qa1 --test “/bin/id”
```

Depending on the role definition and the user’s role assignment, the command might display information similar to this:

```
Testing: User = qa1 command = /bin/id

User qa1 can run the command as 'root' via dzdo, authentication
will not be required, noexec mode is off
```

Using command rights in a standard shell

After command rights have been defined, added to role definitions, and assigned, users can execute privileged commands in a standard shell environment by invoking the `dzdo` command then typing the command to execute, including any command-line options they are allowed to use.

For example, assume you have defines a command right for `shutdown -r` that enables users to execute the command as the `root` user. If you add that right to a role definition—such as the UNIX Login role—that allows users to log on using a standard shell environment, users assigned to that role can execute the command by typing the following:

```
dzdo shutdown -r
```

Using command rights in a restricted shell environment

After command rights have been defined, added to role definitions, and assigned, users can execute commands in a restricted shell environment by typing the command, including any command-line options they are allowed to use.

For example, assume you have defines a command right for `shutdown -r` that enables users to execute the command as the `root` user. If you add that right to a role definition that forces users into a restricted shell environment, users assigned to that role can execute the command by typing the following:

```
shutdown -r
```

Users can only execute the specific command rights that have been added to the role within the restricted shell environment.

Running unauthorized commands

When users are assigned to roles that require a restricted shell environment, the `dzsh` shell provides the subset of commands the user is allowed to run and automatically runs each allowed command as the user the command is configured to run as. If the user attempts to run a command he is not authorized to use in his current role, the shell displays a warning. For example, if the user is not authorized to run the `uname` command in the `dzsh` shell, the following message is displayed:

```
$ uname
uname: command not allowed
```

Setting or changing the active role

Users who are only assigned to one or more restricted shell environments roles are only allowed to run commands within the `dzsh` shell. Within the restricted shell, a user can only be in one active role at a time to prevent ambiguity about the commands the user can run or the user account that should be used to execute those commands.

For example, if the user `carol` is assigned to the `lab_staff` restricted shell environment role that specifies the `tar` command should run as `root` and to the temps restricted shell environment role that specifies the tar command should run as tmp_admin, she needs to specify which role she is using to run the tar commands under the proper account.

Within the restricted shell, users can switch between available roles, as needed, using the built-in role command. If a user has been assigned to the backup_ops role and the dev_managers role, he can run the role command to specify which role should be active so that only commands from that role apply. For example, to switch from the backup_ops role to the dev_managers role:

```
$ role dev_managers
Role changed to: dev_managers
```

For more information about using the role option in a restricted shell, see the man page for dzsh.

Viewing available roles

The dzinfo command enables users to view information about the roles they have available and what they are allowed to do within their different roles. You may want to add this command to all of your restricted environment roles to allow users to check their definitions and availability within the authentication and privilege elevation restricted environment shell.

For more information about using the dzinfo command, see the man page for dzinfo.

Using a graphical desktop manager in a restricted environment

In some operating environments, users who are placed into a restricted environment may not be able to log on using a graphical user interface desktop manager unless they are explicitly given permission to run the desktop manager or related commands within the dzsh restricted environment. For example, on Red Hat Linux, users must be allowed to run /usr/bin/dbus-launch to log on using KDE or Gnome desktop manager.

To allow restricted environment users to log on using KDE or Gnome on Red Hat, you must add dbus-launch to the list of allowed commands for the restricted environment user's role. If you want to prevent restricted environment users from logging on using the graphical user interface, you can restrict their access to specific PAM-enabled applications such as ssh.

Defining rights to use PAM applications

As discussed in Basic concepts of access rights and roles, access rights allow users to perform specific operations. You define the most basic rights—such as the right to log on or connect remotely—when you define roles. To use the rights associated with a role, however, you must be able to authenticate your identity through a pluggable authentication module (PAM) application, such as `login` or `ssh`. This chapter describes how to define PAM application rights that authorize users to log on or access services on Centrify-managed computers.

How applications determine access rights

Most of the programs you run on Linux and UNIX computers are configured to use a pluggable authentication module (PAM) to control access. For example, the `login`, secure shell (`ssh`), and file transfer (`ftp`) services are all PAM-enabled programs. These programs check the local PAM configuration to determine whether a user is allowed to use the requested service.

When you install the Centrify agent and join a domain, you replace the default PAM authentication service with a PAM service that looks for the users and groups to allow or deny access to in Active Directory. Because the PAM service is the first "gatekeeper" to access on most computers, users must have at least one PAM access right to log on at all.

Default PAM access rights

By default, Access Manager creates three predefined PAM access rights in every parent and child zone:

This PAM right	Grants access to
login-all	All PAM applications on a computer joined to the domain. Adding the `login_all` PAM access right allows users to log on and use any PAM-enabled application. The right uses the wild card (*) character to match all PAM application names and is included by default in the predefined UNIX Login role.
ssh	Secure shell sessions on Debian and Ubuntu 6 and 7. Adding the `ssh` PAM access right allows users to log on remotely using secure shell connections on Debian and Ubuntu computers joined to the domain.
sshd	Secure shell sessions on all Linux and UNIX computers except Debian and Ubuntu 6 and 7. Adding the `sshd` PAM access right allows users to log on remotely using secure shell connections on all other distributions of Linux and UNIX computers joined to the domain.

Adding specific PAM access rights

PAM access rights control who can access specific PAM-enabled applications in the zone where they are created and any child zones of that zone. You can add as many **PAM Access** rights as you need to identify the specific PAM-enabled applications users can access. For example, you can add PAM access rights to control who can use file transfer protocol (`ftp`) services on specific computers.

If you want to grant rights to specific PAM applications, however, you must know the appropriate application name on the specific computers you support. For example, if you want to allow Active Directory users to log on and use a default shell, you might create a PAM access right for the `login` program and for a graphical desktop manager such as `gdm`.

What to do before creating a new access right

Before creating a new PAM access right, you should review the operating system of the computers in the zone where you plan to create the new right. The application name might be different on computers with different operating systems. If you are creating separate rights for individual PAM applications, keep in mind that users must have at least one PAM access right or they will not be able to log on to any computers.

Rights required for this task

You can create new PAM access rights if you have been delegated the "Manage roles and rights" administrative task in the Zone Delegation Wizard. If you have not been delegated this task, your user account must be a domain user with the following permissions:

Select this target object	To apply these permissions
Authorization	Click the **Properties** tab, then select **Allow** for the following properties: ▪ Write msDS-AzApplicationData
msDS-OpObjectContainer This object is listed under a globally unique identifier (GUID) for the Authorization object.	On the **Object** tab, select **Allow** to apply the following permissions to this object: ▪ Create msDS-AzOperation objects Click the **Properties** tab, then select **Allow** for the following properties: ▪ Read objectClass

Who should perform this task

In most cases, a UNIX administrator or a delegated zone administrator familiar with PAM applications and the operating system of the managed computers performs this task, depending on your organization's policies.

How often you should perform this task

It is common to add new PAM access rights over time as the need arises and as you develop more granular control over the specific rights different users should be granted.

Steps for completing this task

The following instructions illustrate how to add a PAM access right using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new PAM access right.
3. Expand Authorization, then expand **UNIX Right Definitions**.
4. Select PAM Access, right-click, then click **Add PAM Access Right**.
5. Type a name for the access right.

   The name of the access right can be the same as the PAM application name, or any name that is easily identifiable.

6. Type the name of the PAM-enabled application for which you want to create an access right.

   You can use wildcards to perform pattern matching for the application name. For example, you can specify `*ftp*` to match all PAM-enabled applications containing the string `ftp`, such as `vsftpd`, `ftpd`, and `ftp`.

   The Application Name field supports `glob` pattern matching syntax. For example, the name can contain a question mark (?) to represent any single character, an asterisk (*) to represent any string, including an empty string, or an expression enclosed by brackets (`[. . .]`). For more detailed information about using wildcard patterns and `glob` syntax, see the `glob` man page.

   You should note that application names vary depending on the local operating system where the application is accessed. For example, the following table lists several common PAM-enabled applications and the appropriate application name to use on different platforms.

For this application	On	Use this name
telnet	Common Linux platforms, such as Red Hat, Debian, SuSE, Centos, and Ubuntu, HP-UX, and Irix	`login`
	Sun Solaris	`telnet`
	VMware ESX, Oracle Linux, Scientific Linux	`remote`
ftp	Common Linux platforms, such as Red Hat, Oracle Linux, and Scientific Linux, and VMware ESX	`vsftpd`
	Some Linux platforms, such as Debian, Centos, and Ubuntu, Sun Solaris, HP-UX, Irix	`ftp`

For this application	On	Use this name
graphical desktop	Common Linux platforms, such as Red Hat, Debian, Oracle Linux, Centos, Scientific Linux, and Ubuntu	`gdm`
	Sun Solaris and HP-UX	`dtlogin`
	SuSE and Irix	`xdm`
ssh	Most platforms	`sshd`
	Debain and Ubuntu	`ssh`

7. Type an optional description of the access right.
8. Click **OK** to save the PAM access right.

What to do next

After you define a new PAM access right, you might want to create a new role definition and add this right to it in the current zone or in a child zone. You must add the right to a role to test its operation.

Modifying an existing PAM access right

After you have created and tested a new PAM access right, you might want to modify the right name and description, or the pattern used to match the application name. For example, if you add computers with a different operating system to the zone where the PAM access right is defined, you might have to modify the application name to use wild card characters.

To modify a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to modify, right-click, then click **Properties**.
5. Change the right name, application name, or description for the access right, then click **OK**.

Copying a PAM access right

You should keep in mind that PAM access rights are specific to the zone where you create them. They can be added to any roles you define for the zone or to roles defined in any child zone of the zone. After you define PAM access rights in a zone, however, you can also copy and paste or drag and drop the rights from one zone to another, as needed.

To copy a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to copy, right-click, then click **Copy**.
5. Navigate to the PAM Access node in the new zone, right-click, then click **Paste**.

Deleting a PAM access right

If you are no longer using a specific PAM access right in any roles, you might want to delete the right. For example, if you create new rights for testing purposes and found some of them were not appropriate or failed to work as expected, you might want to delete the rights you aren’t going to use.

To delete a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to delete, right-click, then click **Delete**.

• • • • • •

Renaming a PAM access right

If you only need to change the name of a PAM access right, you can rename the right at any time without modifying the description or the pattern used to match the application name.

To rename a PAM access right using Access Manager:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the PAM access right.
3. Expand Authorization, UNIX Right Definitions, and **PAM Access**.
4. Select the PAM access right to rename, right-click, then click **Rename** and type a new name for the access right.

Using PAM-enabled applications

The default `UNIX Login` role includes the predefined `login-all` PAM access right to enable users to log on and access any PAM-enabled application. If you define specific PAM access rights, users who are assigned a role with that right can only access the specifically authorized PAM-enabled applications. For example, users who are assigned to a role that includes the right to access FTP (`ftpd`) can connect to the FTP server by typing a command similar to the following:

```
ftp ginger.ajax.org
```

Requiring multi-factor authentication for PAM applications

If you select the “Multi-factor authentication required” system right in a role definition, the PAM applications you add to the role will require users to select a secondary form of authentication to log on successfully. You define the forms of authentication available and presented to the user in the authentication profile you have configured using the administrative portal for the Centrify identity platform. For example, you might configure an authentication profile that require users to answer a phone call, click a link in an email message, or respond to a text message.

Note that some applications do not support multi-factor authentication and users might be denied access to applications that they would otherwise be able to use. For example, if a specific version of an application that you want to use only supports a single layer of authentication—such as a password challenge—users would be prevented from logging on and using the service even if they are assigned to a role with the predefined `login-all` PAM application right.

If you want to grant access to applications that only support one layer of authentication in roles where you are generally using the “Multi-factor authentication required” system right, you must add those applications to the list of applications for which you want to skip multi-factor authentication. You can update the list of applications for which to skip multi-factor authentication by enabling and modifying the “Specify programs for which multi-factor authentication is ignored” group policy or setting the `pam.mfa.program.ignore` configuration parameter in the `centrifydc.conf` file.

Before assigning roles with multi-factor authentication required to users, you should test access to all of the applications you expect users to access to verify they won’t be unexpectedly denied access simply because multi-factor authentication isn’t supported. Because the applications that don’t support multi-factor authentication will depend on the platforms and the versions of the applications you plan to support, testing in your own environment is the only way to determine which applications to add to the `pam.mfa.program.ignore` configuration parameter.

The most common applications that are known to only support a single password challenge and response for authentication are ignored for multi-factor authentication by default. For example, some versions of `java` and `vsftpd` do not support multi-factor authentication and are ignored by default.

Additionally, while some platforms support multi-factor authentication for all PAM applications, they may not allow you to require multi-factor authentication for GUI log in. For example, for users running AIX, Solaris, and HP-UX, multi-factor authentication for GUI login is not supported.

Using secure shell session-based rights

As discussed in Default secure shell (SSH) access rights, **Access Manager includes predefined secure shell rights that enable you to** identify the specific secure shell services that a user who has the PAM SSH access right can run. This chapter describes how to use the secure shell (`ssh`) session-based rights that control the operations specific users or groups can perform on Centrify-managed computers.

Secure shell rights require Centrify OpenSSH

SSH has become the defacto standard for administrators and users to securely access remote UNIX systems. The combination of the latest versions of OpenSSH supporting Kerberized connections, along with the Centrify DirectControl Agent directly integrating the UNIX computer with Active Directory's Kerberos infrastructure, provides the administrator with the ideal environment for secured single sign-on. Users logging in from Windows computers can securely access remote UNIX computers using their Active Directory credentials to automatically log in to the UNIX computer.

While many UNIX systems might have an sshd server installed, most are older implementations of the sshd server that do not support Kerberos and newer versions might not have been compiled with support for Kerberos. The Centrify package contains OpenSSH compiled with support for Kerberos by dynamically linking to the Centrify Kerberos libraries to ensure that single sign-on works seamlessly as expected in an Active Directory environment.

This provides several advantages, including:

- The OpenSSH client and server are preconfigured to automatically support PAM and Kerberos.

- There is no need for DNS-to-realm mapping because DirectControl knows the relationship between hosts and their SPNs.
- There is no need for a .k5login file in the user's home directory since DirectControl can automatically map the UPN (User Principal Name) in the Kerberos ticket to the UNIX profile for the Active Directory username presented in the ticket.
- OpenSSH in combination with DirectControl accepts connections to any of the computer's valid hostnames, either fully qualified or not, because all combinations are registered with Active Directory. This further reduces the dependency on accurate DNS entries to enable Kerberos to operate properly.
- The installation process automatically updates the `$PATH`.

The Centrify version of OpenSSH is a separate package that can be installed with the Centrify agent. Before you configure any specific secure shell rights to include in roles, verify that you have the Centrify OpenSSH package installed on your managed computers. The default secure shell rights are only applicable for the Centrify-compiled version of OpenSSH. If you did not select the OpenSSH package as part of a custom installation when you installed the agent, re-run the installation script to install the package before attempting to use secure shell rights.

Secure shell rights require PAM access rights

Before you configure any specific secure shell rights, you should also identify the PAM access right to use. The predefined PAM access right `sshd`—or `ssh` for Ubuntu computers—grants users permission to log on and use all secure shell services on Centrify-managed computers. You must grant the `sshd`, `ssh`, `login-all`, or a custom PAM access right before you can use any secure shell (SSH) rights to restrict access to specific services.

The SSH access rights only work in conjunction with the PAM access right that allows a user to log on using a secure shell session. If a user is not assigned to a role that grants the PAM access right to log on using a secure shell, SSH rights are ignored.

When a user attempts to log on using a secure shell session, `adclient` first verifies that at least one role in effect for the user has the PAM access right that allows him too log on using SSH. If a PAM access right is in effect, `adclient` checks to see which specific SSH rights the user has before allowing or denying the action the user is attempting.

Combining secure shell rights

You can add predefined SSH rights to any role that can be assigned to Active Directory users and can combine different rights for fine-grain control over the specific secure shell operations users are allowed to perform. For Linux and UNIX computers, only the following predefined secure shell session-based rights are available:

- `dzssh-all` grants access to all secure shell services.
- `dzssh-direct-tcpip` allows local and dynamic port forwarding (`ssh-L`, `ssh -D`).
- `dzssh-exec` allows command execution.
- `dzssh-scp` allows secure copy (scp) operations.
- `dzssh-sftp` allows secure file transfer (sftp) operations.
- `dzssh-shell` allows secure terminal (tty/pty) connections.
- `dzssh-Subsystem` allows an external subsystem except `sftp` subsystem which has its own right.
- `dzssh-tcpip-forward` allows remote port forwarding (`ssh -R`).
- `dzssh-tunnel` allows tunnel device forwarding.
- `dzssh-X11-forwarding` allows X11 forwarding.

When combining rights into role definitions, you should keep in mind that some secure shell operations require you to explicitly include the `dzssh-exec` right. For example, if you include the `dzssh-scp` right in a role definition, a user might attempt to execute an arbitrary program with a command line similar to following:

```
ssh troll@localhost scp -S/home/troll/script " -f "
```

Because this command line presents a potential security risk, the operation is not allowed. To prevent the `dzssh-scp` right from being used on its own to execute an arbitrary program on a remote computer, the `-S` command line option is only supported if you also include the `dzssh-exec` right in the role definition. Similarly, you must explicitly include the `dzssh-exec` right in a role definition if you want to support using the `dzssh-sftp` right with the `-S` command line option. For security reasons, only the `dzssh-exec` right allows the remote execution of a program on a target computer.

If the `dzssh-exec` right is not included in the role definition when it is required, users will see an "`access denied`" message.

You should note that you cannot add any secure shell rights to role definitions that allow local users. You can only include them in role definitions for Active Directory users.

Configuring secure shell settings

You can use Centrify group policies to manage several aspects of secure shell (`ssh`) authentication and operation. The Centrify group policies for secure shell are located in the **SSH Settings** folder after you add the `centrify_unix_settings.xml` administrative template to a Group Policy Object. When you enable and configure secure shell group policies, the changes are recorded in the secure shell configuration file, `/etc/centrifydc/ssh/sshd_config`, at the next group policy update interval. To have your changes take effect immediately, run the `adgpupdate` command.

Centrify puts all of the configuration files for secure shell operations in the `/etc/centrifydc/ssh` directory. Depending on your operating system, you might also have other `ssh` configuration files stored in the other locations. When users start a secure shell session and use their secure shell rights, the Centrify agent first checks the `/etc/centrifydc/ssh` directory for configuration files, then looks for configuration file in the `/usr/local/etc` directory on AIX computers, and in `/etc/ssh` directory on most other Linux and UNIX computers.

At a minimum, you should enable the **Enable application rights** group policy in a Group Policy Object that applies to the site, domain, or organizational unit that contains Centrify-managed Linux and UNIX computers.

To configure the secure shell group policy for application rights

1. On a Windows computer, open the Group Policy Management console.
2. Select an appropriate Group Policy Object, right-click, then select Edit.

You can select any Group Policy Object that applies to the site, domain, or organizational unit that contains Centrify-managed Linux and UNIX computers.

3. Expand Computer Configuration > Policies > Centrify Settings > SSH Settings and double-click **Enable application rights**.

4. Click Enable, then click OK.

   This setting adds the following parameter to the `/etc/centrifydc/ssh/sshd_config` file:

   ```
   ServiceAuthLocation /usr/share/centrifydc/libexec/dzsshchk
   ```

   This parameter sets the path to the `dzsshchk` command. The `dzsshchk` command verifies the access rights for users when they log in with SSH for all computers to which the group policy object applies.

You can also use secure shell group policies to control other configuration settings, such as the allowed and denied groups and users and authentication processing. For example, you can use the following group policies to configure operations for Centrify OpenSSH connections:

- **Add sshd_config properties** enables you configure secure shell properties defined in the `sshd_config` file by group policy. If you enable this group policy, you can add and edit properties as name-value pairs.
- **Allow challenge-response authentication** enables you use multi-factor authentication if you are using the secure shell package installed with the operating system. This group policy is not required if you are using the Centrify OpenSSH package for the agent.
- **Allow groups** specifies the list of groups whose members are allowed to log on through `sshd`.
- **Allow GSSAPI authentication** enables authentication either as the result of a successful key exchange, or through GSSAPI user authentication.
- **Allow GSSAPI key exchange** enables authentication using a key exchange based on GSSAPI.
- **Allow users** specifies the list of users who are allowed to log on through `sshd`.
- **Deny groups** specifies the list of groups whose members are not allowed to log on through `sshd`.
- **Deny users** specifies the list of users who are not allowed to log on through `sshd`.

- **Enable application rights** allows secure shell applications to grant secure shell rights.
- **Enable PAM authentication** to use PAM account and session handling.
- **Permit root login** specifies whether the root account can be used to log in using `ssh`.
- **Set banner path** specifies the path to a local file that is sent to a remote user requesting authentication.
- **Specify authorized keyfile** specifies the file that contains the public keys that can be used for user authentication.
- **Specify ciphers allowed for protocol version 2** enables you to add or delete ciphers allowed for single sign-on connections.
- **Specify client alive interval** specifies a timeout interval, in seconds, for requesting a response to client alive messages.
- **Specify log level** specifies the level of detail to record in the log file for messages from `sshd`.
- **Specify login grace period** specifies the time, in seconds, after which the server disconnects if a user has failed to log in.
- **Specify maximum client alive count** specifies the maximum number of client alive messages that may be sent by the secure shell daemon (`sshd`) without receiving a response from the client.

For more information about adding administrative templates for group policies to a Group Policy Object and how to configure and apply the group policies for secure shell, see the*Group Policy Guide*.

Configuring secure shell parameters

The following parameters apply to specific usage for SSH.

`ServiceAuthLocation`

Uncomment this line in `sshd_config` to enable the SSH application right feature. Refer to the pre-defined `scp`, `sftp`, and `winscp` for how to utilize this feature. Default is `disable`.

`AuditSshCommandline`

Set this parameter in `sshd_config` to `yes` if the command line options are displayed in the audit trail message. Default is `no`.

`krb5ccUnique`

Set this parameter to `yes` to specify when storing the Kerberos credentials cache. Centrify `sshd` generates a unique credential cache name for it. If this parameter is set to no, the old style credential cache name, `krb5cc_<uid> or KCM:<uid>`, is used.

`SSOMFA`

Set this parameter is yes to support Single Sign-On (SSO) with Multi- Factor Authentication (MFA). The MFA order is determined by the `AuthenticationMethods` keyword in `sshd_config`. This keyword works only when `UsePAM` is enabled and the Centrify keyword `ServiceAuthLocation` is set. Default is no.

Note: MFA is not supported for authentication using public key.

Creating and assigning custom role definitions

Access rights and role definitions are intended to give you maximum flexibility to grant or restrict access for the users and groups in your organization. This chapter describes how you can configure role-based access controls for Centrify-managed computers by adding custom access rights to custom role definitions and assigning those custom role definitions to users and groups. This chapter provides examples to illustrate how you can create and assign custom role definitions. The authorization scenarios you can support might be far more complex than the examples described in this guide, however.

Reviewing the fundamentals of role definitions

As discussed in Basic concepts of access rights and roles, rights are fundamental to authorizing user access, you cannot assign rights directly to users. Instead, rights are combined into **role definitions** that reflect the needs of a specific job function, such as database administrator, or the ability to perform a particular task, such as start a web service or run commands that compress or extract files. It is up to you, as an administrator, to decide on the role definitions your organization needs and to assign those custom role definitions to the appropriate users and groups.

Basic access rights require Active Directory users to have a complete UNIX profile and at least one role assignment, for example by using the UNIX Login role, that is in effect in the zone to which a computer is joined. To move beyond basic access rights, you must define custom rights and custom role definitions, then add the specific rights to each role definition.

After you configure a role definition with rights, you can assign it to individual Active Directory users or to Active Directory groups, so that the role applies to all members of the group. By assigning role definitions to groups, you can manage ongoing role-based user access completely through Active Directory.

Combining rights into role definitions

Rights can be combined in a variety of ways to accomplish different goals. In general, however, role definitions fall into one of these broad categories:

- Roles that grant access to one or more PAM applications and a standard UNIX shell.

  With this type of role, Active Directory users can log on using all or a specified PAM application, such as `login` or `ftp`, and execute commands that are commonly available to non-administrative users. This type of role can only be assigned to Active Directory users or groups.

- Roles that grant users additional privileges to execute administrative commands and perform administrative tasks they would not be able to perform with a standard user account.

  With this type of role, users can temporarily elevated their privileges to execute administrative commands by first invoking the `dzdo` command, which is similar to `sudo`. This type of role can be assigned to Active Directory users or to local users.

- Roles that provide access to a specific subset of shell commands in a customized restricted environment shell (`dzsh`).

  With this type of role, users can execute the commands explicitly defined for them in a restricted shell environment. This type of role can be assigned to Active Directory users or to local users.

In preparing role definitions for different groups of users, you should keep in mind that the rights from multiple role assignments accumulate. For example, you could use one role definition to control login rights, and another role definition to specify a set of privileged commands. By separating login rights from privileged access rights, not every role definition requires PAM application or UNIX system rights.

Creating a root-equivalent role definition

Most organizations require at least one `root` user role definition that is equivalent to specifying `ALL:ALL` in a `sudoers` file or giving users access to the `root` password on their computers. The purpose of this role definition is to allow selected users to execute privileged commands on a regular basis. The role definition allows them to execute commands without being given the `root` password or having privileges hard-coded in individual `sudoers` files on multiple computers.

Because this role definition enables system administrators to execute privileged commands without the `root` password, you can improve security for the organization and reduce the chance of an audit finding for access to the `root` password.

You can create this role definition in a parent zone or a child zone to control its scope. In most cases, you should only assign the role in a child zone or on an individual computers.

Define the right for running all commands

Rights and roles are defined at the zone level and inherited down the zone hierarchy. If you define a right in the top-level zone, it is available in all child zones. If you define a right in a child zone, it can be used in that zone and any of its child zones. Similarly, you can define roles in the top-level parent or any child zone, depending on where you want to make the role available. In this example, the right to run all commands as the `root` user is defined in a top-level parent zone.

The following instructions illustrate how to define a right for running all commands using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To define a right for running all commands as root:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.

   For this example, select the top-level parent zone so that this command right is available in all child zones.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a

description for this right, then define the right to run all commands like this:

- Type an asterisk (*) in the Command field to indicate all commands are allowed.
- Select Specific path and type an asterisk (*) in the field to indicate that any path is allowed.

6. Click the Restricted Shell tab and deselect the **Can be used in a restricted role** option if you want to prevent this command from being used in a role that uses a restricted shell environment.
7. Click the Run As tab to verify the command can be used with `dzdo` and is set to run as `root` by default.
8. Click **OK** to use the default environment variable settings and command attributes.

   Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Create a role definition for running all commands

After you have defined the right to allow a user to run any command with `root` privileges, you can create a role definition for that right. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

To create a role definition with the right to run all commands as root:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role, then click **OK**.

For example, type a name such as `root_equivalent` and descriptive text such as `Users with this role can run any command with root privileges`.

Optionally, you can select **Allow local accounts to be assigned to this role** if you want to assign both Active Directory users and local users to the role. This option is only available when you first create a role definition. You can also click **Available Times** if you want to limit when the role is available for use. By default, roles are available at all times.

If you using the `UNIX Login` role to grant access to computers in the zone and want to use the default auditing level of **Audit if possible**, you can click **OK** then skip to Step 8.

6. If you are not assigning the `UNIX Login` role to grant access to computers, click the System Rights tab and select the following options:

   - Password login and non-password (SSO) login are allowed
   - Non-password (SSO) login is allowed
   - Login with non-Restricted Shell

   Note that you cannot set these system rights if you selected the option to allow local users to be assigned to this role.

7. If you don't want to use the default auditing level, click the Audit tab.

   - Select **Audit not requested/required** if you have the auditing service enabled but don't want to audit user activity when this role is used.
   - Select **Audit if possible** to audit user activity where you have the auditing service enabled.
   - Select **Audit required** to always audit user activity. If the auditing service is not available, users in this role are not allowed to log on.

8. Select the new role definition, right-click, then click **Add Right**.

9. Select the right you defined for running all commands as `root`, then click **OK**.

Assign an Active Directory group to the role

You should associate Centrify role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership. For example, create an

Active Directory group named sanfrancisco_role_rootequivalent. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for root-level access, such as root_equivalent, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role.
   - Select Group as the object to find.
   - Optionally, type all or part of the group name.
   - Click Find Now,

   Select the group you created for the role in the results, then click **OK**.
7. Click **OK** to complete the assignment.

Creating a role definition for a shared service account

The root-equivalent role definition provides centralized management for a limited number of administrators who have permission to execute all commands on selected computers. Another common reason for defining a role is to execute privileged commands associated with a service account. In many organizations, service account passwords are known by multiple users, making them a security risk. For example, all of the database administrators in the organization might know the password for an oracle service account, an account with permission to perform privileged database operations. Because the password is shared information, it presents a security risk and a potential audit finding that might have costly consequences.

Setting up a role definition for a service account involves creating a command right for switching to the service account user and defining a PAM access right for role.

Define the right for switching to a service account

The steps for defining a right for switching to the service account user are similar to defining the rights for the root-equivalent user, but the definition is more restrictive.

To define a right for switching to a service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name for this command right and, optionally, a description for this right, then define the right to switch to the service account.For example, if the service account is `oracle`:
   - Type `su - oracle` in the Command field.
   - Verify the Standard user path is selected.
6. Click the Restricted Shell tab, under Can be used in a restricted role, select **Specific user or uid**, then type `root`.
7. Click the Run As tab, deselect **Can be used by dzdo**.

   These settings specify that this right can only be used in a restricted shell environment and users can only run the commands that are explicitly allowed in the restricted role they are assigned. If this is the only right defined for a role, the only command users assigned to the role can run is `su - oracle`. For a role definition with this right to be effective, you would add command rights for the specific database operations users should be allowed to perform after switching to the `oracle` service account. For example, if the `oracle` service account is used to run a `backup-all-dbs` script, you would add a right to allow the execution of that script.
8. Click **OK** to use the default environment variable settings and command

attributes.

Alternatively, you can click the Environment and Attributes tabs if you want to view or set additional properties for this right definition.

Define a PAM access right to allow logging on

The default `UNIX Login` role allows users to log on using a password or without a password in an unrestricted environment. If you are creating a role for a service account, you can use PAM access rights to control the specific commands users can use to log in. To illustrate controlling how users log on, this example of a restricted role for the `oracle` service account only allows users to log on with `ssh`.

To define a PAM access right for a specific application:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new PAM access right.
3. Expand Authorization > UNIX Right Definitions.
4. Select PAM Access, right-click, then click **Add PAM Access Right**.
5. Type a name and, optionally, a description of the PAM application for which you are adding an access right.

   For the Application field, type the platform-specific name for the PAM application as defined in the PAM configuration file or PAM directory. For example, type `ssh` or `sshd`. You can also use wildcards in this field to perform pattern matching for the application name.
6. Click **OK** to save the access right for this PAM-enabled application.

Create a restricted role definition for the service account

After you have defined the rights that allow a user to log on using a PAM-enabled application and run the `su -` command for a service account, you can create a role definition for these rights. You must create a role definition somewhere in the zone hierarchy before you can assign users to the role.

• • • • • •

To create a restricted role definition for switching to a shared service account:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role, then click **OK**.

   For example, type a name such as `oracle_service` and descriptive text such as `Users with this role can start a secure shell session and switch to oracle`.

   By default, this role is available at all times. You can click **Available Times** if you want to specify days of the week or select times of the day for making the role available.
6. Click the System Rights tab and select at least one option that allow users assigned to this role definition to log on, then click **OK**.

   In this example, users open a secure shell to switch to the service account so you might select **Non-password (SSO) login is allowed**.

   If a service account instead of a user account is used to log on, it might be mapped to a disabled Active Directory account. In this case, you might select the **Account disabled in AD can be used by sudo, cron etc** system right to ignore the disabled state and allow the service account to log on.
7. Select the new role definition, right-click, then click **Add Right**.
8. Select the rights you defined for running the switch user (`su -`) command and logging on with the PAM application `ssh`, then click **OK**.

Assign an Active Directory group to the role

You should associate Centrify role definitions with Active Directory security groups so that you can manage them using the processes and procedures you have for managing Active Directory group membership. For example, create an Active Directory group named `sanfrancisco_role_oracle`. You can then assign the new role definition to that group.

To assign the role definition to an Active Directory group:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to assign the role definition.
3. Expand Authorization.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for using secure shell and switching to the service account access, such as `oracle_service`, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory security group you created for the role definition.

   - Select Group as the object to find.
   - Optionally, type all or part of the group name.
   - Click Find Now.

   Select the group you created for the role in the results, then click **OK**.
7. Click **OK** to complete the assignment.

Working in a restricted shell environment

When users who are assigned to this role want to open a secure shell session and switch to the `oracle` service account, they will be placed in a restricted shell environment. Within the restricted shell, they can only execute the commands you have added to the role definition until they exit the restricted shell session. In this example, the role definition only allows users to log on using `ssh` and execute one command, `su - oracle`. If those users are also assigned the `UNIX Login` role, they will have access to an unrestricted shell when they close the restricted shell session.

If you want users who access a shared service account to work exclusively within the restricted shell environment, you must remove the `UNIX Login` role assignment in the zone or on the computer where they should only have restricted shell access. Before removing the `UNIX Login` role assignment, however, you should consider the trade-off between improved operational security and audit compliance and reduced operational access. Depending on the rights you add to a role that runs in a restricted shell environment, the restricted shell can dramatically limit what users can do.

Testing access in a restricted shell

If you create a role definition for a shared service account that runs in a restricted shell environment, you should test it before migrating any users to it. You can use the `dzinfo` command with the `--test` option from a UNIX command prompt. For example, type `dzinfo`, the user name to test, the `--test` option, then the full path to the command to test:

```
dzinfo raejames --test "/usr/bin/su - oracle
```

You can also run the `dzinfo` command with the `--roles` option to see information about the rights defined for the current user or a specified user. For example, run the following command to check the roles and rights defined for the user `raejames`:

```
dzinfo raejames --roles
```

For more information about using this command, see the `dzinfo` man page.

What users see in a restricted shell environment

For users assigned to a role that runs in a restricted shell, logging on opens a `dzsh` shell. Within that shell users can only execute the commands you have explicitly defined for them. In this example scenario for a shared service account, typing `su - oracle` is the only allowed command. If the user types any other command, the shell reports that the command is not allowed.

Creating a role definition for temporary root access

Another common use case for role definitions occurs when you want to provide temporary access to privileged commands. For example, you might want to provide temporary root-level access to an application developer troubleshooting a problem on a production server or to a consultant you've hired for a specific period of time. These types of role definitions are often used as overrides on individual computers.

The steps for creating a role definition with temporary root access are similar to the steps for creating the other roles, except that you specify time constraints for the role. The time constraints might include specific hours of the day, days of the week, or a start and end time for a role assignment. The next sections summarize the steps for creating a role with temporary root-level access.

• • • • • •

Define a command that allows root access

The steps for defining a right for switching to the `root` user are similar to defining the right to run commands for the root-equivalent user, but Centrify recommends you create a separate right definition for this case.

To create the right to switch to the root user:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name, such as `emergency_access`, for this command right and, optionally, a description for this right, then define the right to switch to the `root` user:
   - Type the command for switching to the `root` user. For example, type `su - root` in the Command field.
   - Verify Standard user path is selected.
6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

   These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.
7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

   In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.

Create a role definition for temporarily running as root

After you have defined the right to switch to the `root` user, you can create a role definition for that right.

To create a role definition with the right to run the emergency_access command:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role.

   For example, type a name such as `emergency_access` and descriptive text such as `Users with this role can temporarily run commands with root privileges`.
6. Click **Available Times** to specify days of the week or select times of the day for making the role definition available.

   For example, you might want to allow access only on Friday, Saturday, and Sunday and deny access the rest of the week. After you have set the days and times for the role definition to be available, click **OK**.
7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select the `emergency_access` command you defined for switching to the `root` user, then click **OK**.

   To use this role, a user must be assigned to the `UNIX Login` role for the zone or a role definition that has at least one UNIX system right, such as Password login and non-password (SSO) login are allowed.

Assign the role as a computer-level override

In most cases, a role definition of this type is assigned to a specific computer rather than applied to all computers in a zone.

To make a role assignment on an individual computer:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer for which you want to define a computer-level role assignment.
3. Expand Computers, then select the specific computer on which you want to make a role assignment.
4. Select Role Assignments, right-click, then click **Assign Role**.
5. Select the role definition you created for temporary root access, such as `emergency_access`, then click **OK**.
6. Click **Add AD Account** to search for and select the Active Directory user who should have temporary root access:
    - Leave User as the object to find.
    - Optionally, type all or part of the use name.
    - Click Find Now.

   Select the user in the results, then click **OK**.
7. Deselect **Start immediately** and set a specific Start time for the role assignment.
8. Deselect **Never expire** and set a specific End time for the role assignment.
9. Click **OK**.

Verify the role assignment on the computer

You can run `dzinfo --roles` or `dzinfo username --roles` to see if the `emergency_access` role is available based on the start time for the role definition and the local time of the Linux or UNIX computer.

At the specified start time for the role assignment on the local computer, the user you assigned to the `emergency_access` role can type the following command:

```
dzdo su - root
```

The user is not prompted to provide the password and becomes the `root` user on the local computer until the specified role assignment end time. The one caveat to be aware of is that the user would continue to have `root` access after the specified end time if the shell session remains open continuously. If a user is still logged on after the time period has expired, you should check whether the user still requires `root`-level access. If the session has remained open but the user should no longer have `root` access, kill the session and log the user off.

Creating a role definition with specific privileges

The previous examples of role definitions granted broad privileges. You can also use role definitions grant or deny very specific rights. For example, you might want to deny access to a specific set of commands for a specific group of administrators who otherwise have broad access rights or to strictly limit exactly what commands users can execute. Depending on the requirements of your organization, you might configure these types of role definitions to be used in a restricted or unrestricted shell.

The steps for creating a role definition with specific privileges are similar to the steps for creating the other roles. In this example, rights are defined to prevent the execution of specific commands and combined with a right to grant access to all commands not explicitly listed.

Define command rights to prevent the use of commands

The steps for defining rights that deny access to specific commands are similar to the steps defining other rights, but require different syntax. In this example, you create a “blacklist” of commands users cannot execute.

To create the right to switch to the root user:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new command right.
3. Expand Authorization > UNIX Right Definitions.
4. Select Commands, right-click, then click **New Command**.
5. On the General tab, type a name, such as `No password resets`, for this command right and, optionally, a description for this right, then define the right:

- Type `!passwd *` in the Command field.
- Verify Standard user path is selected.

An exclamation point (!) at the start of a command disallows matching commands. Command rights that start with the exclamation point take precedence over others that don't.

6. Click the Restricted Shell tab and verify **Can be used in a restricted role** and **User running the command are selected**.

   These options enable you to use this command right in combination with other rights in a role definition that requires a restricted shell environment.

7. Click the Run As tab and verify **Can be used by dzdo** and **Any user** are selected, then click **OK**.

   In most cases, you can leave the default settings for the other properties. If you want to make changes, click the Environment and Attributes tabs before saving the new command.

8. Repeat Step 4 to Step 7 to create rights for the following specific commands:

   ```
   !groupadd *
   !useradd *
   !groupdel *
   !userdel *
   ```

Create a restricted shell role definition that uses the command rights

After you have defined all of the command rights that disallow specific commands, you can create one or more role definitions to use those rights. For example, you might create one role definition to run in an unrestricted shell that requires users to invoke dzdo to execute privileged commands and another role definition that runs in a restricted shell but does not require users to execute privileged commands using dzdo. The second role might be useful if you have existing scripts that would have to be modified if invoking dzdo is required.

To create a role definition for specific command rights:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name where you want to create the new role definition.
3. Expand Authorization.
4. Select Role Definitions, right-click, then click **Add Role**.
5. Type a name and description for the new role.

   For example, type a name such as `operators` and descriptive text such as `Users with this role can run privileged commands but not reset passwords, add or delete users and groups`.
6. Click **System Rights** if you want this role definition to be used in a restricted shell environment as a replacement for the predefined `UNIX Login` role.

   To use this role, a user must be assigned to a role definition that has at least one login system right, such as Password login and non-password (SSO) login are allowed or Non-password (SSO) login is allowed.
7. Click **OK** to save the role definition.
8. Select the new role definition, right-click, then click **Add Right**.
9. Select all of the command right that disallow specific operations, the command right that grants access to all remaining commands, and a PAM access right, then click **OK**.

   For example, you might add the following previously-defined command rights to this role definition:

   ```
   No password resets
   No user adds
   No group adds
   No user deletes
   No group deletes
   Root like access (* for all commands not explicitly
   disallowed)
   PAM ssh/login allowed
   ```

   This role definition allows members of the `operators` role to execute any command within a restricted shell environment except those explicitly disallowed, including privileged commands, without invoking dzdo first. You can assign the role definition to the appropriate Active Directory users or groups like the previous role definitions.

Create an unrestricted shell role definition that uses the command rights

The command rights were configured to allow execution in either a restricted shell environment or an unrestricted shell environment. In an unrestricted shell environment—for example, the default shell environment when users are assigned the `UNIX Login` role—commands that require administrative privileges must be executed by first invoking the dzdo command, which is similar to invoking commands with sudo.

You can control whether users are required to enter a password or another form of authentication when they execute privileged commands using dzdo by setting one of the **Re-authenticate** options on the Attributes tab when you create a command right. By default, no password is required. If you were adding a new command right that requires re-authentication, you would click the Attributes tab, then select **Re-authenticate current user** or **Re-authenticate using target user's password**. For more information about these options, see Requiring re-authentication to run commands.

In most cases, the default of no password is appropriate because the user has been previous authenticated before invoking dzdo to execute a privileged command and the **Re-authenticate using target user's password** option requires the user to know the privileged account password. For example, if select this option and the run-as user is `root`, the user must know the password for the `root` account.

The steps for creating the role definition that includes the previously-defined command right are the same for the unrestricted shell as for the restricted shell except that, at Step 6 in the topic Create a restricted role definition for the service account, in the System Rights tab you would also select the **Login with non-Restricted Shell** option if you are not using the `UNIX Login` role. You could add all of the same command rights to the role definition and grant the same privileges and exceptions.

The primary difference between the two role definitions would be how users execute their privileged commands.

In the restricted shell environment, users running the `adflush` command requiring administrative privileges:

```
dzsh $ adflush
```

In the unrestricted shell environment, users running the `adflush` command requiring administrative privileges:

```
[tulo@ajax]$ dzdo adflush
```

Creating a role definition with rescue rights

The Rescue rights option allows you to control which users should be able to log on if problems with authentication, the authorization cache, or the auditing service are preventing all other users from logging on. For example, if you have a computer with sensitive information, such as credit card numbers or intellectual property, you might require auditing for all users in the role with access that computer. If the auditing service is stopped or removed on that computer, no one would be able to log on and use the computer until auditing is restored. If you create a role with the Rescue rights option selected, only the users assigned to that role are able to log on and continue working until the problem that caused the lockout is found and fixed.

Users who are in a role granted access because they have rescue rights can still be audited through the system logging facility. However, their activity is not recorded in the audit store database if the auditing service is not available.

Creating a role definition that allows local users

Most role definitions are only applicable to Active Directory users and groups. In some cases, however, you might want to create a role definition that can be assigned to local users. For example, you might want to assign local users to a role that grants rescue rights to ensure a specific local account can log on if an Active Directory user is not available.

Role definitions that allow local users to be assigned cannot include PAM access rights or SSH rights, however, and therefore do not include any of the UNIX system rights. You can use role definitions that allow local users to assign specific command rights to local and Active Directory users. You can also set the audit level for the role definitions that allow local users to be assigned.

If you select the option to allow local users, you can specify the local accounts when you assign the role by clicking **Add Local Account**, then typing the name of local UNIX or Windows accounts to assign to the role. The **Add Local Account** option is not displayed when assigning a role definition that does not allow local accounts.

Creating a role definition for secure shell rights

You can add SSH rights to any role definition as long as the role does not accept local users. Although SSH rights require the PAM `ssh` right, the role definition to which you add SSH rights does not require the PAM access right. As long as a user is assigned to a role that includes the PAM `ssh` right, you can add SSH rights to any other role definition to make the rights effective.

In addition to adding the rights to a role definition, you must set the `ServiceAuthLocation` parameter in the `sshd_config` configuration file to check for secure shell rights when users log on using a secure shell. In most cases, you should use the **Enable application rights** group policy to set this parameter for all Centrify-managed Linux and UNIX computers. This group policy sets the path to the `dzsshchk` command which verifies the specific applications rights for users when they log on.

Alternatively, you can manually set this parameter on an individual computer by editing the configuration file to include the following:

```
ServiceAuthLocation /usr/share/centrifydc/libexec/dzsshchk
```

Creating additional custom roles and role assignments

The previous sections described common role definitions that organizations implement to begin the process of migrating and removing locally defined privileged accounts. For most organizations, locally-defined accounts with privileged access present a security risk and are often identified as a compliance issue by auditors.

By creating role definitions similar to those described in this chapter, you can eliminate the need to share `root` and service account passwords while still providing privileged access to computers where it’s needed. These additional roles are not required, however. You can choose to create them or create a completely different set of role definitions to suit your organization. For example, you might decide to create custom roles specifically tailored to the needs of database administrators, backup operators, and web application developers. Similarly, you might decide to create separate role definitions that are customized with AIX command rights for AIX administrators that are different from the command rights defined for Solaris administrators.

As with the common role definitions, additional custom role definitions can be created in the top-level parent zone and available throughout the zone hierarchy or in any child zone. They can also span all the computers in a zone or be assigned specifically to individual computers.

If you plan to create your own custom role definitions and role assignments, keep the following key points in mind:

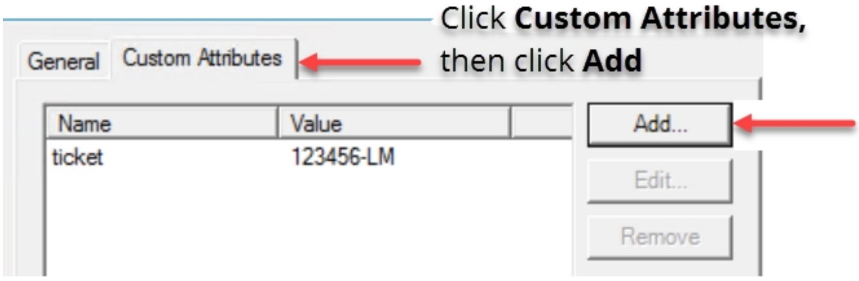
- Rights associated with roles are cumulative. Users receive all of the rights in all of the roles they are assigned.
- Users must be assigned at least one role that allows an interactive login or Kerberos authentication to have any access to any computers. For existing users, this is accomplished by assigning the default `UNIX Login` role during the migration to Active Directory.
- Users must be given the Login with non-Restricted Shell system right to have access to a full shell. If they are in a role without this right, they can only execute the commands explicitly defined for their role.

  For users who have previously had full shell access, this limitation can be frustrating, unexpected, and unworkable. Before placing or moving users into a restricted role, be sure those users and managers throughout the organization are well-informed and well-prepared for the change and understand the business reasons for the change.

Adding custom attributes

You can add custom attributes to role definitions and role assignments. For example, you might want to use a custom attribute to reference a ticket number associated with a specific type of access request, role definition, or temporary role assignment. Custom attributes are optional and you can use them to capture any kind of information that is meaningful to your organization.

You can add custom attributes when defining or modifying a role, defining or modifying a computer role, or when modifying role assignment properties.



Click **Custom Attributes**, then click **Add**

Name	Value
ticket	123456-LM

Exporting and importing rights and roles

You can export rights and role definitions from any zone if you want to save part or all of the information to a file. You can then import all or part of that information into a new zone and modify it, if needed. For example, you can choose to export all the rights you have defined in one zone but create a completely new set of role definitions for those rights in another zone. Exporting and importing provides a convenient way to copy and paste multiple rights and role definitions at one time.

Rights, roles, and role assignments are all inherited from parent to child zones, so there is no need to import or export any authorization information within a zone hierarchy. However, if you have multiple parent zones—for example, representing different geographical regions—you might want to use export and import to copy authorization information from one geographical region's zone to another.

Exporting authorization information

You can export multiple rights and role definitions to an `.xml` file that you can then use to import these definitions into another zone. You can also copy and paste or drag and drop individual rights and role definitions between zones.

To export rights and role definitions:

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name from which you want to export authorization information.

3. Select the Authorization node, right-click, then click **Export Roles and Rights**.
4. Select the information you want to export, then click **Next**.

   For example, select **All** to export all of the rights and all role definitions. Selecting all or individual role definitions exports all of the rights included in those role definitions.
5. Click **Browse** to specify a location and file name for the export file, then click **Next**.
6. Review the information to be exported, then click **Finish**.

Importing authorization information

You can import multiple rights and role definitions that you have previously exported from a zone and saved to an `.xml` file. You can also copy and paste or drag and drop individual rights and role definitions between zones.

To import rights and role definitions

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you want to import authorization information.
3. Select the Authorization node, right-click, then click **Import Roles and Rights**.
4. Click **Browse** to navigate to the file that contains the authorization information you want to import, then click **Next**.
5. Select the information you want to import, then click **Next**.
6. Review the information to be imported, then click **Finish**.

Updating rights, roles, and role assignments

When you make changes to rights, roles, or role assignments, these changes take effect on managed computers at the next cache update interval, as set by the `adclient.cache.expires` parameter or the “Set object expiration” group policy. The default update interval is 10 minutes. If you want changes to take place immediately, you can flush the cache on individual computers.

To flush the cache and update authorization changes:

1. Log on or switch to the `root` user on the managed computer.
2. Run the `adflush` command to clear the agent cache. For example:

   ```
   /usr/sbin/adflush
   ```

Working with computer roles

In previous chapters a *role definition* described a specific set of access rights for a user or group, including the period of time when those access rights were in effect. A *computer role* is an association that enables you to make the most of those role definitions. Role definitions grant specific access rights or enforce certain access restrictions. Computer roles enable you to associate role definitions with computers that share a similar function or have a common attribute. This chapter describes how you can configure and use computer roles to manage access rights for different sets of users.

How computer roles provide flexibility

Centrify-managed computers can only be joined to one zone at any time. This limitation makes it difficult to manage granular access rights at the zone level alone. Computer roles enable you to group computers that share a common function or attribute and associate the group of computers with a specific set of role assignments to users or groups. Individual computers can be members of any number of computer roles with different sets of users who have different access rights based on their role assignments.

Computer roles can have multiple role assignments

A computer role associates a group of computers with a set of role assignments. For example, you might have several computers that host Oracle database instances. Using a computer role, you can associate the group of computers that host an Oracle database with one role assignment that grants some users full administrative access. That same computer role can associate the same group of computers with a second role assignment that grants some users access to specific commands that must be run using the `oracle` account.

That same computer role can also associate the same group of computers with a third role assignment that grants application users permission to log on using a secure shell session. As long as the set of computers remains the same, you can use the same computer role to grant different sets of users different access rights.

Managing access using multiple computer roles

Computer roles enable you to manage access rights using multiple filters. For example, you might have several computers that host Oracle database instances. Some of the computers that host an Oracle database might also belong to specific departments, such as the finance or engineering organizations. Some of the computers that host an Oracle database might run Red Hat Enterprise Linux while others have a Solaris operating system. You can use computer roles to grant different sets of access rights based on the criteria you want to use to group the computers. In this example, you might have one computer role for Oracle database servers and their database administrators, another computer role for users in the finance and engineering departments, and another computer role for IT staff who specialize in managing either Linux or Solaris computers.

Computer roles enable you to define access rights using any grouping criteria that makes sense for your organization. In this case, you might have one computer role linked with the Active Directory security group for all Oracle servers, a second computer role linked with the security group that only has computers that belong to the finance or engineering organization, and a third computer role linked with the security group for Linux or Solaris computers. If the set of computers grouped together changes, you should use a new computer role to grant different sets of users different access rights.

Planning to use computer roles

Because computer roles provide you with a great deal of flexibility for defining access rights, you might want to do some planning before you create new computer roles. For example, before you create a computer role you must know the criteria you want to use to group computers into one or more Active Directory security groups. You must also identify the users who will have a common set of access rights based on the computer grouping.

At a high-level, defining a computer role requires the following:

- Identify a unique Active Directory security group for each computer role.

  You should identify an attribute the computers in a particular group share, such as computers in the web farm, that host specific applications, or serve a specific department. You can create the group and add computers to it in Access Manager when you create the computer role, or before creating the computer role using Active Directory Users and Computers.

- Identify the sets of users that share common access rights and create Active Directory groups for them.

  You might want to define multiple sets of user-based roles. For example, a computer role for Oracle servers might require a “database users” group, a “database administrators” group, and a “backup operators” group.

- Identify the access rights and role definitions for each set of user-based roles.

  You might want to create specific rights, role definitions, and role assignments for different sets of users, or use existing roles. For example, the “database users” group might only require the predefined UNIX Login role definition, while the “database administrators” group might require access to privileged commands, and the “backup operators” groups might be only be allowed to run a specific set of commands in a restricted shell.

Creating a new computer role

A computer role is similar to a zone in that it defines a group of computers, a set of users, and specific access rights for a combination of computers and users. However, computer roles do not require a computer to be joined to the zone where the computer role is defined and a computer can be a member of multiple security groups and thus multiple computer roles.

Because computer role assignments define a relationship between a security group of computers, a set of rights in a role definition, and a security group of users, they control who can do what on specific computers. You can change the list of computers or the list of users dynamically simply by changing the security group membership.

What to do before creating a new computer role

Before you create computer roles, you must join a domain and zone. You should also decide on the criteria to use for grouping computers. Each computer might belong to several different security groups to used in different computer roles. Depending on your organization's policies for creating security groups, you might want to prepare one or more Active Directory security groups for Centrify-managed computers.

Rights required for this task

To create computer roles, your user account must be a domain user with the following permissions:

Select this target object	To apply these permissions
msDS-AzScope This object is listed under a globally unique identifier (GUID) for the Authorization object. For example: `CN=cab186af-61a0-4d54-a0dd...`	Click the **Properties** tab and select **Allow** to apply the following properties to this object only: ▪ Read description ▪ Read msDS-AzScopeName ▪ Read msDS-AzApplicationData ▪ Write description ▪ Write msDS-AzScopeName ▪ Write msDS-AzApplicationData

Who should perform this task

A UNIX zone administrator or a Windows domain administrator who is responsible for adding and maintaining security groups performs this task, depending on your organization's policies.

How often you should perform this task

It is common to create new computer roles any time you identify new criteria for grouping computers and role assignments.

Steps for completing this task

The following instructions illustrate how to create a new computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To create a new computer role using Access Manager

1. Open Access Manager.
2. Expand **Zones** and the individual parent or child zones required to select the zone name that will contain the new computer role.
3. Expand **Authorization to select Computer Roles**, right-click, then click **Create Computer Role**.
4. Type a name for the computer role and an optional description, then select either **<Create group>** to create a new Active Directory group for computers or **<...>** to search for an existing group of computers to use.

   For example, click **Create group** to create a new Active Directory security group named `oracle_servers` for the computers that host Oracle database instances. If creating a new group, you are prompted for the location, group name, and scope.
5. After you have selected or created an Active Directory security group, click **OK**, then click **OK** to save the new computer role.

   Note: If you're using classic zones, you cannot add cross-forest groups to roles at this time. All groups added to roles should be defined in the local forest. However, users from a trusted forest may be added to groups in the local forest and then added to a role, or they may be directly added to a role. (Ref: IN-90001)

Adding computers to a computer role

After you create an Active Directory security group for computers and associate it with a computer role, you can add or remove computers simply by updating the group membership. For example, if you have a computer role for managing access to Oracle database servers and you deploy a new instance, you simply add the new server to the computer security group you created for Oracle servers. You can update the group membership using Active Directory Users and Computers, Access Manager, ADEdit, or another tool of your choice.

After you have specified the Active Directory security group you want associated with a computer role, the account membership is synchronized so you can use Access Manager or another program to make changes.

Steps for completing this task

The following instructions illustrate how to add computers to a computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To add computers to the computer role using Access Manager

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer role to which you want to add computers.
3. Expand Authorization and Computer Roles, then expand the computer role to which you want to add computers.
4. Select Members, right-click, then select **Add Computer**.
5. Type all or part of a computer name, then click **Find Now** to search for the computer accounts to add.
6. Select one or more computers from the results, then click **OK** to automatically add computers to the Active Directory group associated with the computer role.

Adding role assignments to a computer role

For computer roles to be effective, you must create the access rights and role definitions for different sets of users. You can then assign the appropriate predefined or custom roles to different sets of users to grant or restrict their rights within the scope of the computer role. With proper role definitions and role assignments, you can manage access rights for computers completely through group membership. For example, after you have created the role definition for Oracle database administrators, you can add and remove group members to the group you created for Oracle administrators in Active Directory.

For information about creating access rights and role definitions, see the following:

- Defining rights to run privileged commands
- Defining a restricted shell command right
- Adding specific PAM access rights
- Combining secure shell rights
- Creating and assigning custom role definitions

After you have create the appropriate access rights and role definitions, you must assign those roles to the appropriate users and groups to complete the configuration of the computer role.

Steps for completing this task

The following instructions illustrate how to add role assignments to a computer role using Access Manager. Examples of scripts that use the Access Module for Windows PowerShell, ADEdit, or the Centrify Windows API are available in other guides, the *Centrify Software Developer's Kit*, or in community forums on the Centrify website.

To associate user role assignments with a computer role using Access Manager

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that contains the computer role to which you want to add role assignments.
3. Expand Authorization and Computer Roles, then expand the computer role to which you want to add role assignments.

4. Select Role Assignments, right-click, then select **Assign Role**.
5. Select the role definition that you want to add to the computer role, then click **OK**.
6. Click **Add AD Account** to search for and select an Active Directory user or security group to assign to the role.

   You can select User or Group as the object to find, type all or part of the user or group name, then click **Find Now**. For example, type “ora” to search for and select the “oracle_db_admins” Active Directory group then click **OK**.
7. Click **OK** to complete the role assignment for the selected user or group in the selected computer role.

Repeat these steps for each role definition you want to assign to users and groups in this computer role. For example, if you have an Active Directory “oracle_db_users” group that should be allowed to log on and run shell commands on the computers in the “oracle_servers” computer role, you would select the predefined UNIX Login role in Step 5 and assign that role definition for the computers in the “oracle_servers” computer role to the “oracle_db_users” group in Step 6.

Viewing and modifying a computer role

You can view information about computer roles by expanding Authorization and Computer Roles for a zone. However, computer roles are also closely linked to the Active Directory groups that define their scope and role assignments, so there are several different ways you might view or modify information about a computer role. For example, you might use Access Manager, Active Directory Users and Computers, or ADEdit commands, depending on what you are trying to do.

In Access Manager, you can expand a computer role, then select **Role Assignments** to see the users, groups, and role definitions that have been assigned on the computers that are members of the computer role. You can also expand a computer role, then select **Members** to see the computers to which the role assignments apply. To see the Active Directory group assigned to the computer role in Access Manager, select the computer role, right-click, then select **Properties**.

If you are using Active Directory Users and Computers, you can view the properties for the Active Directory group associated with the computer role and click the **Members** tab to see the computers assigned to the computer role.

If you want to add a computer to an existing computer role, you can simply add that computer to the Active Directory group associated with the computer role without making any changes in Access Manager. Similarly, if users join or leave your organization, you can simply add or remove those user accounts in the appropriate Active Directory groups that are associated with the computer role. For example, if you define the `oracle_servers` computer role to associate a specific set of computers with a role assignment that grants administrative rights to users in the Active Directory security group `oracle_db_admins,` you could simply add the user account for `Frank.Smith` to the Active Directory security group `oracle_db_admins` to give that user administrative access on the computers that are members of the `oracle_servers` computer role. You do not need to make any changes in Access Manager.

To modify the rights and role assignments for a computer role, you must use Access Manager or ADEdit commands.

Using computer roles

Deciding how best to use computer roles requires some planning and configuration that might not be part of your initial deployment plan. To make effective use of computer roles, you must also prepare appropriate role definitions for different sets of users. However, computer roles provide a powerful and flexible option for managing access to computers using your existing processes and procedures for managing Active Directory group membership.

After you create a computer role, it is easy to manage even as your organization changes and grows. For example, if another Oracle database server comes online, you add it to the computer group you created for Oracle database servers in Active Directory. If other DBAs join your organization, you add them to the Active Directory group you created for Oracle administrators. The computer role links the computer group to the role assignment and no additional updates are needed to accommodate these kinds of organizational changes. If you need to modify the access rights, you can change the role definition and have the changes apply to all members of the group.

Requiring multi-factor authentication using computer roles

Computer roles enable you to group and provide access to computers through role assignments. One strategy you might find useful is to use computer roles to control where multi-factor authentication should apply. For example, you might have several computers with highly sensitive material where you want to ensure all user access will require multi-factor authentication. To accomplish this goal, you can configure a computer role, then add and remove computers with sensitive information to control whether multi-factor authentication is required.

To require multi-factor authentication based on a computer role

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name that will contain the new computer role.
3. Expand Authorization to select Computer Roles, right-click, then click **Create Computer Role**.
4. Type the role name and, optionally, a role description, then select **<Create group>** for the Computer group to create a new Active Directory group for computers.

   For example, to create a new Active Directory security group for the computers with sensitive information, click **Browse** to select the Active Directory location for the new group. If you are using the default deployment structure, you would browse to a location similar to `centrify.pubs.org/Centrify/Computer Roles` then type a group name such as `mfa_required_servers`, select a scope, and click **OK**.
5. Click **OK** to save the new computer role.
6. Add the computers that require multi-factor authentication for access to the `mfa_required_servers` Active Directory security group.

   As you add computers to the Active Directory security group, the computers are listed as Members of the computer role.
7. Expand the computer role you creates in Step 4, select Role Assignments,

right-click, then select **Assign Role**.

For example, if you created a new computer role with the role name `CR_MFA_required`, expand that computer role name to select Role Assignments, right-click, then select **Assign Role**.

8. Select the predefined `require MFA for login` role definition, then click **OK**.

9. Select **All Active Directory accounts**, then click **OK**.

Working with managed computers

This chapter describes how to add Linux and UNIX computers to Active Directory domains, manage computer accounts and properties, perform common administrative tasks, and leave the domain.

Identifying who can add computers to the domain

Who can join computers to a domain depends on your organization's policies and those policies are enforced through Active Directory. In general, there are two common scenarios:

- Any authenticated domain user can add up to ten computers to the domain.

  This is the default behavior for Windows computers. Many organizations follow this policy, so that administrative access is not required to add computers to a domain.

- Only users with specific permissions can add computers to the domain.

  Some organization restrict who can add computers to the domain. For example, a user might have to be a member of the Domain Admins or Account Operators group to add computers to a domain.

The policy your organization follows for Windows also applies when you want to add Linux and UNIX computers to a domain. If any authenticated user can add a Windows computer to the domain, adding a Linux or UNIX computer does not require an administrative user name and password. If only administrative or delegated users are allowed to add computers to the domain, the user adding a Linux or UNIX computer must provide an administrative or delegated user name and password.

If you aren’t sure whether an administrative account is required to join a domain, you can prepare computer account before attempting to join the domain, and allow the computer account itself to be used to join the domain. Performing this type of “self-service” join simplifies the operation and allows the computer account to manage its own password without administrative intervention.

Preparing computer accounts before joining

If joining the domain is restricted to privileged users, or if you know that you will need to specify computer-level overrides, you can prepare computer accounts in advance for the Linux and UNIX computers you want to add to the domain.

There are several advantages to preparing computer accounts before joining the domain. For example, preparing a computer account enables you to accomplish the following:

- Specify the user, group, or computer account with permission to join the computer to the domain.
- Define the organizational structure you want to use for computers in Active Directory.
- Delegate administrative tasks for managing the computer account.
- Specify the user or group with permission to manage computer-level overrides for the computer.

By preparing the computer account in advance, you can minimize the changes or configuration steps you might otherwise have to perform after joining the domain. For example, by identifying the account to use when a computer joins the domain you can ensure users can add their own workstations without being assigned any special rights. By selecting the appropriate organizational unit for the computer account ahead of time, you minimize the need to move the computer account after joining the domain.

To prepare a computer account using Access Manager:

1. Open Access Manager.
2. Expand Zones and any parent or child zones required to select the zone name to which you want to add the computer account.
3. Right-click, then click **Prepare UNIX Computer**.

4. Select the type of preparation you want to perform, then click **Next**.

   In most cases, you should select both options to ensure the appropriate user or group has the permissions required to join the domain and set computer-level overrides.

5. Choose whether to create a new computer object or select an existing computer object, then click **Next**.

   If the computer account exists, but you want to add a zone profile and delegate permission to join the domain and manage computer overrides for the computer, click **Browse** to search for and select the existing computer object. After selecting an existing computer account, click **Next** to continue to Step 7.

6. Type the computer name to use for the new computer account and specify a location for the computer account object in Active Directory, then click **Next**.

   - For **Computer name**, type the host name to use for the computer account in Active Directory.
   - For **Domain**. verify the domain name displayed is the appropriate domain for the computer account to join. Click **Browse** to navigate to a different Active Directory domain.
   - For **DNS name**, verify the DNS name for the computer account. You can modify the DNS name for the computer, if needed. For example, if computer names in DNS use a different suffix than the Active Directory domain, you might need to modify the default value displayed.
   - Select **Create the computer object in the container** to specify the parent container for the new computer account in Active Directory. In most cases, you should use the default parent container object. Click **Change** to navigate to a different container object for the computer account.

7. Select the **Allow this computer to join the domain using a read-only domain controller** option if you want the computer to join itself to the domain using a read-only domain controller and select the type of license to use, then click **Next**.

   If you click **Next** without selecting **Allow this computer to join the domain using a read-only domain controller**, the computer must join the domain by connecting to a writable domain controller.

8. Review the default list of service types and service principal names for the specified computer, then click **Next** to accept the default set of service principal names.

   If you want to make changes to the default services or service principal names, you can do the following:

   - Click **Add** to add a service type or add a new service name to an existing service type.
   - Select a service principal name and click **Edit** to change the name.
   - Select a service principal name and click **Remove** to delete the name.
   - Click **Default SPN** to restore the default list of service principal names.

   If you are in an environment where multiple instances of the same SPN are possible, as a user with administrator privileges, use the `-d` or `--forceDeleteObjWithDupSpn` parameter with the `adjoin` command to ensure duplicate SPNs are removed.

9. Select whether to allow a specific user or group to join the computer to the domain or use the computer account and automatically-generate password to join the domain, then click **Next**.

   In most cases, select **Allow the computer to join itself to the zone** to allow the computer account to perform a “self-service” join. This option is selected by default because it allows you to automate the join operation so that a user name and password are not required.

   If you want a specific user, group, or computer account to be used to join the domain, select **Allow this user, group, or computer to join the computer to the zone** then click **Browse** to search for the user, group, or computer that you want to give permission to join the computer to the domain.

10. Select the user, group, or computer account with permission to set computer-level overrides, then click **Next.**

    By default, the permissions required to manage computer-level overrides are granted to members of the Domain Admins group. You can click **Browse** to search for and select another user, group, or computer account.

11. Review your configuration settings, then click **Next**.

12. Review the confirmation of the operation performed, then click **Finish**.

The computer account is created in Active Directory and a zone profile for the computer is added to Access Manager in the zone's Computers container. The user or group you have designated as the trustee can now join this computer to the domain using the `adjoin --selfserve` command line option, and the group you designated for computer-level overrides can add users and role assignments to the computer.

Delegating permissions when preparing a computer account

When you prepare a computer account, you have the option to grant a specific user, group, or computer account the administrative permissions required to perform two separate tasks:

- The permissions required to join the computer account to the domain.
- The permissions required to set and manage computer-level overrides

In most cases, you should select both options even if you want to grant different accounts the permissions required to perform each task.

However, it is possible to create a computer account and not delegate permission for computer-level overrides by deselecting the **Delegate permission for machine overrides** option. If you deselect this option, you are the only administrator who can set profile or role assignment overrides for the computer. No other user or group will be granted the permissions required to set or manage computer-level override for user profiles or role assignments.

Likewise, it is possible to delegate permissions for computer-level overrides without preparing the computer to join the domain by deselecting the **Prepare computer for adjoin** option. If you deselect this option, the computer icon appears in the zone, but the Active Directory computer object and service connection point are not created. The designated trustee can set computer-level override for user profiles or role assignments. No other user, group, or computer account will be specifically granted the permissions required to join the domain.

If any authenticated user can add computers to the domain, then any user with a valid domain account can join Linux and UNIX computers to the domain. If adding computers to a domain requires an administrative account, only the administrator who creates the computer account can join it to Active Directory. For more information about who can add computers to a domain, see Identifying who can add computers to the domain.

Allowing password resets for computer accounts

If you use Access Manager and the Prepare UNIX Computer wizard to create a computer account before joining the domain, you can select the **Allow the computer to join itself to the zone** option to set the permissions required for a computer to manage its own account. If you use Active Directory Users and Computers to create a computer account, however, you need to manually modify the permissions for the account.

By default, most computer accounts do not have permission to reset their own account password. This prevents the delegation of administrative rights for the computer to the local computer account. If you want to give a computer account administrative rights in a zone, you need to modify the computer account to allow password resets. In addition, allowing a computer account to update its own properties enables Access Manager to display the agent version and maintain operating system information for the computer account.

Checking for the appropriate permissions

To check whether a computer account allows password resets, you can view the permission settings for the account.

To check and modify the permissions for a computer account:

1. Open Active Directory Users and Computers, expand the domain, and select Computers to find the computer account to which you want to assign administrative rights.
2. Select the computer account, right click, then select **AD Properties**.
3. Click the **Security** tab, scroll down the list of group or user names and select **SELF**.
4. In the list of Permissions for SELF, scroll to the **Reset Password** permission, click **Allow**, then click **OK**.
5. Select the computer account, right-click and select **Reset Account**, then click **Yes**. When the account is reset, click **OK**.

Assigning administrative rights to computer accounts

After you have checked the Active Directory permissions for a managed computer account and modified them, if necessary, you can assign zone administrative rights to the account through Access Manager.

To give administrative rights to the computer account:

1. Open the Access Manager console.
2. In the console tree, select **Zones**, and if necessary, **Child Zones**, then select and expand the zone in which you are interested.
3. Right-click, then click **Delegate Zone Control**.
4. Click **Add**, select **Computer** from the Find list, then click **Find Now**.
5. In the results, select **Domain Computers**, click **OK**, then click **Next**.
6. Click **Join computers to the zone** and optionally, **Remove computers from the zone**, then click **Next**.

   Note: In most cases, these are the only administrative tasks you should assign to the computer account. You can, however, give the account additional rights, if needed. For information about the permissions associated with each delegated task, see the*Planning and Deployment Guide*.

7. Click **Finish**.

Joining a domain

To begin authenticating users and authorizing access to Linux and UNIX computers and resources, you must first add the computers you want to manage to the appropriate Active Directory domains in one or more Active Directory forests. You can do this by running the `adjoin` command interactively or by using the `adjoin` command in a script. A successful join operation is what converts a Linux or UNIX computer into a **Centrify-managed computer**.

Connecting to the domain controller

To add a new computer to a domain, you must specify the domain you want to join. The `adjoin` program then locates an appropriate domain controller for the domain you specify and connects to Active Directory through that domain controller. By default, the domain controller to contact is determined by the Active Directory site topology. If the nearest domain controller in the site is not available, the agent attempts to connect to the next closest domain controller in the site. If no domain controller can be contacted or the connection takes too long to complete, the join operation fails.

If you don't want to agent to select a domain controller based on the site topology, you can specify a master domain controller on a zone-by-zone basis. If you specify a master domain controller, the agent will connect to the appropriate domain controller based on the zone you are joining.

What happens during the join operation

If the Centrify agent can successfully connect to an Active Directory domain controller, it performs a series of key tasks to complete the join operation. For example, during the join operation, the `adjoin` program completes the following tasks:

- Starts the Centrify UNIX agent `adclient` process.
- Checks whether a computer account already exists for the local computer in Active Directory. It creates a new Active Directory computer account for the local computer, if needed.
- Sets the password on the Active Directory computer account to a randomly-generated password. The password is encrypted and stored locally on the UNIX host to ensure that only the Centrify agent has control of the account.
- Updates the Kerberos service principal names used by the host computer, generating new a Kerberos configuration file and `krb5.keytab` entries, and generating new service keys for the `host` and `http` services.
- Synchronizes the local computer's time with Active Directory to ensure the timestamps for Kerberos tickets are accepted for authentication.

After joining a domain

By default, computers function exactly the same after joining the domain as they did before joining the domain. Local users can continue to log on and existing programs and applications can continue to work as they did before joining the domain. The primary difference after joining the domain is that you have more complete control over access to the computer and what Active Directory users who are granted access can do. You will also have more tools at your disposal for managing computer properties and operations. For example, after joining a domain, you can use any combination of the following tools:

- Access Manager
- Access Module for Windows PowerShell
- ADEdit command line programs and scripts
- Active Directory Users and Computers
- Group Policy Management console and Centrify group policies

You can use any of these tools to add Active Directory users to the appropriate zones, and to define and assign appropriate rights and roles for the users who need access to Linux and UNIX computers.

Joining a domain and zone with the adjoin command

In most cases, you add a computer to the domain by running the `adjoin` command directly on a local computer. You run this command once for each Linux or UNIX computer you want to add to a domain in the forest. Using the administrator or a designated user account, you can run the command interactively at the command line or include the command in a script to automate joining a domain.

Specifying the most common arguments

Whether you join the domain interactively from the command line or using a script, you must specify a few required arguments. You might also need to specify several additional arguments, such as a user name and password for an account with permission to join the domain, an alias for the computer in Active Directory, or the organizational unit in which to place the computer.

The most common format for the `adjoin` command is:

```
adjoin --user username --zone zonename domain
```

For example, the following command illustrates the most common format for the `adjoin` command:

```
adjoin --user shea@acme.com --zone LinuxDev sales.acme.com
```

This command connects to Active Directory as the user `shea@acme.com` to add the local computer to a previously-created zone called `LinuxDev` zone and to the `sales.acme.com` domain. In this example, the zone and domain name are required. The user name is not a required argument—if not specified the `adjoin` command would prompt for the `Administrator` account password. However, because the user `shea` is a member of the `acme.com` domain rather than the sales.acme.com domain, the user account must be specified in the user_name@domain_name format.

Because the password is not specified in the command line, the adjoin program prompts for the Active Directory password to authenticate the shea@acme.com account before connecting to Active Directory.

In most cases, you should avoid including the password for an account as part of the adjoin command line for security reasons. If you are using adjoin in a script, however, you must include the --password option or provide another mechanism for inputting a valid password. For more information about adjoin command line options and running adjoin commands, see the adjoin man page.

If the adclient process is able to connect to Active Directory and the join is successful, a confirmation message is displayed. By default, the join operation adds the new computer account to Active Directory in the domain_name/Computers container. If the connection to Active Directory fails, a warning message is displayed and the join operation fails.

Using the self-serve option for a previously-created computer account

If you have previously prepared a computer account in Active Directory as described in Preparing computer accounts before joining, you can use the --selfserve (-S) option to join a domain without specifying a user name and password. For example, you can run a command similar to the following to join the domain:

```
adjoin --selfserve domain
```

For example:

```
adjoin --selfserve cendura.org
```

Note that you must specify the domain to join but not the zone—the computer is automatically joined to the zone in which the computer object was pre-created.

If you want to preserve service principal names (SPN) configured in the centrifydc.conf, use the adjoin command option -r spn or --useConf spn. This option only works in conjunction with the -S, --selfserve command.

Joining a domain in workstation mode

In most cases, zones are required if you are adding Linux and UNIX computers to Active Directory to address account migration and role-based access rights.

However, it is possible to deploy without using zones to organize computers, rights, roles, and other information.

The workstation mode is intended for computers that function in the same way as Windows workstations where any valid user can log on to any computer that is joined to the domain. In general, workstations do not require you to manage identity attributes, such as UIDs and GIDs, or access-related attributes, such as the hours a user is allowed to log on. To mirror this behavior for Linux and UNIX computers, the workstation mode automatically creates a local user profile for users when they log on and does not apply any access rules unless you configured them for the user account in Active Directory.

Computers that join the domain using workstation mode are added to a single Auto Zone and are treated the same as Windows workstations, and are managed by Active Directory and group policy settings. You can use the workstation mode and Auto Zone for any computers that do not require profile management or role-based access controls. You can also have any combination of workstation computers that don't require profile management and access control and workstations and servers that do require profile management, access control, hierarchical zones. For more information, see Using workstation mode and Auto Zone.

To join a domain using workstation mode instead of zones, you can run a command similar to the following:

```
adjoin --workstation --user username domain
```

For example:

```
adjoin --workstation --user kai.rodriguez cendura.org
```

This command adds the local computer to a single Auto Zone. The Auto Zone requires no configuration and there are no properties, user profiles, or access rights to manage. All Active Directory users and groups in the forest, or in forests with a two-way trust, can access the computers in the Auto Zone.

Joining the domain using the computer account

On the computer to which you have given administrative rights, run the `adjoin` command and set the `user` name parameter to the computer name with a dollar sign ($) appended and the `password` to the computer name.

```
adjoin domain --zone zoneName --user computername$ --password
computername
```

For example, if the computer name is `valencia` and the Active Directory domain is `arcade.com`, you would run a command similar to the following:

```
adjoin arcade.com --zone finance --user valencia$ --password
valencia
```

Setting the password interval for managed computers

After joining a domain, the password for the managed computer account in Active Directory is automatically reset at a regular interval to ensure security. By default, the password for the computer account is updated with a new, randomly-generated password every seven days. You can customize how frequently the password for the account is changed through the **Password change interval** group policy or by modifying the configuration file, `centrifydc.conf`, on any managed computer.

Allowing a managed computer to authenticate NIS users

If you are using one or more managed computers as a NIS server to provide “agentless” authentication to NIS client requests or to publish NIS network maps, you can identify those computers in Access Manager by setting the **Allow this computer to authenticate NIS users** option on the computer’s Centrify Profile. Setting this option adds the computer account to the `zone_nis_servers` Active Directory group. Additional configuration is required if you want a managed computer to respond to NIS client request.

This option is provided because using Centrify Network Information Service (`adnisd`) is more secure than using a legacy NIS server and can be useful to accommodate certain situations, such as a transition from NIS domains to Active Directory domains. However, continuing to use NIS client requests to retrieve network information is not a secure practice and might result in an audit finding in some organizations.

For more information about installing, configuring, and using the Centrify Network Information Service, see the *Network Information Service Administrator’s Guide*.

Changing the zone for a managed computer

When you join a domain, you must join a specific zone unless you are using workstation mode and connecting through Auto Zone. Over time, you might want to migrate managed computer accounts from one zone to another. You can change the zone information for a computer at any time, if needed. You can change the zone for a computer by selecting a new parent or child zone in the computer properties or by cutting and pasting the computer object from one zone to another in Access Manager.

After you change the zone for a managed computer, you must restart the Centrify UNIX agent on that computer for the change to take effect. You are not required to run `adleave` or `adjoin` to complete the change. For example, after you have changed the zone for a computer log on to that computer and run the following command:

```
/etc/init.d/centrifydc restart
```

Alternatively, you can restart the managed computer to restart all services, including the Centrify UNIX agent. In most cases, however, restarting the agent is sufficient.

> **Note:** If the computer has role assignments defined, you might be prevented from moving the computer until you remove the role assignments.

Changing domain information for a managed computer

Once a computer successfully joins a domain, you can remove it from a domain at any time by using the `adleave` command. You must also use the `adleave` command before you can join a new domain or make changes to the domain information for a computer, such as changing the computer name.

Leaving a domain

Leaving the domain before attempting to join a new domain or changing a computer name ensures that there will not be file conflicts or orphaned information that might prevent the join operation from completing.

You should note that leaving the domain removes all of the Centrify-specific information for the managed computer from Active Directory and reverts any computer settings that were changed by the `adjoin` command to their pre-`adjoin` condition. These changes include reverting PAM, NSS, and Kerberos configuration files to their pre-`adjoin` states and deleting the `/etc/krb5.keytab` file. Leaving the domain does not delete the Active Directory computer object itself.

Leaving the domain does not delete the Active Directory computer object itself. If you want to completely remove any record of the computer from Active Directory, you must delete the computer object using Active Directory Users and Computers.

Joining a different domain

After running the `adleave` command, re-run the `adjoin` command with the appropriate arguments to join a different Active Directory domain. For example:

```
adjoin --zone arcade.com --user gale.harris operations.acme.com
```

For more information about using the `adjoin` and `adleave` commands, see the `adjoin` or `adleave` man page.

Renaming a managed computer

If you need to rename a Linux or UNIX computer that is joined to a domain, you should first leave the domain, rename the computer, then rejoin the domain. Otherwise, you could have issues with the service connection point or service principal name for the computer.

Customizing configuration settings for a computer

You can configure many aspects of the environment for individual computers by applying a Group Policy Object to a site, domain, or organizational unit that includes managed computers and enabling Centrify-specific group policies. For example, you can use policies to customize PAM operations, the length of time to wait for connections between the Centrify UNIX agent and Active Directory, or how frequently to change the computer account password. For information about the group policies available and how to enable them, see the *Group Policy Guide*.

If you are not deploying Centrify group policies, you can also customize the configuration settings in any computer’s local agent configuration file, `centrifydc.conf`. The comments within the file describe the most common settings. For more information about setting the parameters directly in the agent configuration file, see the *Configuration and Tuning Reference Guide*.

Enabling FIPS-compliant encryption

The Federal Information Processing Standard 140-2 (FIPS 140-2) describes US Federal government requirements that IT products should meet for sensitive, but unclassified use. The standard is published by the National Institute of Standards and Technology (NIST) and is required by all non-military agencies of the United States Government. This standard is also widely used by many other organizations outside of the government.

The standard defines the security requirements that must be satisfied by a cryptographic module used to secure unclassified information. There are four levels of security: from Level 1 (lowest) to Level 4 (highest). These levels are intended to cover the wide range of potential applications and environments in which cryptographic modules might be deployed. The security requirements cover areas related to the secure design and implementation of a cryptographic module.

The Centrify agent can be configured to use FIPS-compliant encryption so that a managed computer can successfully join a domain that is FIPS 140-2, Level 1, compliant.

Verifying the Windows environment

Before you configure the Centrify agent to use FIPS-compliant encryption, you should verify that the Active Directory domain meets the minimum requirements for FIPS-compliance. For a Centrify-managed computer to join a FIPS 140-2 Active Directory domain, the Active Directory domain must meet the following basic requirements:

- The domain must be at domain functional level Windows Server 2008, or later.

- The forest must have a global catalog computer that is running at domain functional level Windows Server 2008, or later.
- The domain must have at least one Windows Server 2008 R2, or later, domain controller.
- Any trusted domains you plan to access must be at domain functional level Windows Server 2008, or later.

Although a managed computer can successfully join a domain that has trust relationships to domains at a lower functional level, it cannot access users in those trusted domains, for example, to add user profiles or roles to a zone.

Using group policy for FIPS compliance

If your Active Directory forest meets the minimum requirements and you have configured the Windows environment with the local or group “System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing” security policy, you can make Centrify-managed computers FIPS-compliant by enabling and applying the Centrify “Use FIPS compliant algorithms for encryption, hashing and signing” group policy. You should not use the equivalent Windows group policy to configure FIPS-compliant communications for Linux and UNIX computers. The Centrify “Use FIPS compliant algorithms for encryption, hashing and signing” group policy is specifically designed to support Active Directory domains that are configured for FIPS 140-2 compliance.

The Centrify “Use FIPS compliant algorithms for encryption, hashing and signing” group policy is defined in a separate XML (`centrifydc_fips.xml`) or ADM (`centrifydc_fips.adm`) template file. The template file is included in the Centrify group policy extension. You must add one of these templates to a Group Policy Object to make a Centrify-managed computer FIPS-compliant mode. For information about adding template files and enabling group policies, see the Group Policy Guide. After you enable the policy, it takes effect at the next group policy update interval. To have the policy applied immediately, run the `adgpdupdate` command.

Using the XML template group policy

If you use the XML group policy template to enable FIPS mode, the policy verifies that each computer is joined to a domain at the domain functional level Windows Server 2008, or later. If a domain controller does not meet this minimum domain functional level, the policy issues a warning that allows you to skip enabling of FIPS mode for that computer.

The XML group policy template also verifies *all* computers to which the policy applies are running a supported operating system. On the computers that are running a supported operating system, the policy sets the `fips.mode.enable` configuration parameter to true and automatically stops and restarts the `adclient` process. After the restart, the computers where the policy was applied are FIPS-compliant.

If the computer is *not* running a supported platform, the XML policy leaves the `fips.mode.enable` configuration parameter set to `false`, and does not stop and restart `adclient`. The computer remains joined and the current encryption and hashing algorithms remain in force.

Modifying the agent configuration file

The Centrify “Use FIPS compliant algorithms for encryption, hashing and signing” group policy sets the `fips.mode.enable` parameter in the Centrify configuration file to `true`. By default, this parameter is set to `false` until the group policy is applied and the computer is updated at the next group policy update interval. You can also manually modify this parameter setting directly in the agent configuration file (`centrifydc.conf`), then restart the `adclient` process to enable FIPS mode. In most cases, however, you should use the group policy to set the configuration parameter to enable FIPS mode rather than manually editing the `fips.mode.enable` parameter on individual computers.

Applying the group policy to a domain

In most cases, you should apply the “Use FIPS compliant algorithms for encryption, hashing and signing” group policy to a Windows Server 2008, or later, domain to enable FIPS mode. If the group policy is applied to the domain, then the computer will be enabled for FIPS mode automatically when it joins the domain.

Agent requirements for FIPS-compliant encryption

You can only configure FIPS mode for Centrify agents, version 5.0.2, or later. In addition, FIPS mode is only supported on specific distributions of Linux and Mac OS X operating systems. For a complete and up-to-date list of the platforms that Centrify supports in FIPS mode, see the NIST validation entry for Centrify FIPS mode.

NTLM authentication

The Centrify agent does not support NTLM authentication through SMB or SMB2 when configured to use FIPS-compliant encryption. FIPS mode only allows NTLM pass-through authentication over SChannel. Note that NTLM pass-through authentication requires a Windows Server 2008 R2, or later, domain controller.

Non-compliant operations

When configured to run in FIPS mode, the agent uses non-FIPS compliant *hash* and *key-hash* algorithms, as follows:

- MD4, MD5 and HMAC-MD5 are used to support NTLM passthrough authentication (including using NLTM for PAM authentication).
- MD4 is used to generate the managed computer password hash for use in setting up AES NetLogon Secure Channel. AES NetLogon Secure Channel is used for NTLM pass-through authentication as well as for updating operating system version attributes.
- MD5 is used to generate the UNIX password hash to verify against the MD5 password hash that is stored in the cache during disconnected mode login. (This is for backward compatibility support; this happens when you upgrade from a DirectControl version that does not support the SHA256 password hash.)

When configured to run in FIPS mode, the agent uses a non-FIPS compliant *encryption* algorithm, as follows:

- Non-FIPS compliant encryption will be used in encrypting secret information for internal communication through a UNIX domain socket.
- A non-FIPS compliant random number generator is used in generating the Initialization Vector used in the encryption.

Configuring the encryption types for trusted domains

Inter-realm keys for the `AES256-CTS` and `AES128-CTS` encryption types must be established between any trusted domains to enable Active Directory users from these domains to log on to the joined computer. You can use the `ksetup` utility, installed by default on the domain controller, to set up the inter-realm keys.

To configure the inter-realm keys

1. On the domain controller, open a Command Prompt window.
2. Type the following commands:

   ```
   C:\>ksetup.exe /SetEncTypeAttr trustedDomain AES256-CTS-HMAC-
   SHA1-96
   ```

   ```
   C:\>ksetup.exe /SetEncTypeAttr trustedDomain AES128-CTS-HMAC-
   SHA1-96
   ```

   Note: If you are using pre-validated Active Directory users, you must enable these users for Kerberos AES 128- and 256-bit encryption. You can do so by editing user accounts in Active Directory or by setting attributes for the users in ADSI Edit. For more information, see Enabling required encryption types for pre-validated users.

Manually granting write permissions for a computer account

If the domain that the managed computer is joining does not have at least one Windows Server 2008 R2 domain controller, you must manually grant write permission for the `Operating System Version` and `msDS-supportedEncryptionTypes` attributes to the computer account of the joined computer.

To grant write permission for required attributes to the computer account

1. Open Active Directory Users and Computers or ADSI Edit.
2. Expand the Computers container and select the computer that is joining the domain, right-click, then click **Properties**.
3. Click the Security tab, then click **Advanced**.
4. Click **Add**.
5. In the “Enter the object name to select” field, type `SELF` and click **OK**.
6. Click the Properties tab, select **This object only** from the Apply to list, then

scroll down and click **Allow** for the following attributes:

- `Write msDS-supportedEncryptionTypes`
- `Write Operating System Version`

7. Click **OK** in each dialog box to close the dialog and save the new permissions.

Manually granting write permissions for a user account

If the domain that the managed computer is joining does not have at least one Windows Server 2008 R2 domain controller, you must manually grant write permission for the `Operating System Version` and `msDS-supportedEncryptionTypes` attributes to the user account used to join the computer to the domain.

1. Open Active Directory Users Computers or ADSI Edit.
2. Expand the Computers container and select the computer that is joining the domain, right-click, then click **Properties**.
3. Click the Security tab, then click **Advanced**.
4. Click **Add**.
5. In the “Enter the object name to select” field, type the name of the Active Directory user who will join the computer to the domain and click **OK**.
6. Click the Properties tab, select **This object only** from the Apply to list, then scroll down and click **Allow** for the following attributes:
   - `Write msDS-supportedEncryptionType`
   - `Write Operating System Version` attributes
7. Click **OK** in each dialog box to close the dialog and save the new permissions.

Enabling required encryption types for pre-validated users

If you are using pre-validated Active Directory users, you must enable Kerberos AES 128- and 256-bit encryption for these users. You can do so by editing the user accounts in Active Directory Users and Computers or by setting attributes for the users in ADSI Edit.

To enable encryption for pre-validated users by using Active Directory Users and Computers

1. On the domain controller, open Active Directory Users and Computers.
2. Navigate to the domain and select **Users**.
3. Select the pre-validated user, right-click, then click **Properties**.
4. Click the Account tab, then select the following Account options:
   - This account supports Kerberos AES 128 bit encryption.
   - This account supports Kerberos AES 256 bit encryption.
5. Click **OK** to save the updated account information.

To enable encryption for pre-validated users by using ADSI Edit

1. On the domain controller, open ADSI Edit.
2. Navigate to the domain and select **CN=Users**.
3. Select the user, right-click, then click **Properties**.
4. In the Attribute Editor tab, select the `msDS-supportedEncryptionTypes` attribute and select **Edit**.
5. Type `0X18` to set the hex value for the attribute and click **OK**.

   You should see that the value shows:

   `0x18=(AES128-CTS-HMAC-SHA1-96 | AES256-CTS-HMAC-SHA1-96)`
6. Click **OK** to save the new setting.

How Centrify FIPS mode affects other encryption settings

If you enable FIPS mode, you cannot specify the Data Encryption Standard when joining the domain. The `adjoin --des` option is not supported. Only AES authentication is supported.

If you have specified multiple types of encryption for the computer by setting the `adclient.krb5.permitted.encryption.types` parameter in the `centrifydc.conf` configuration file, only `aes256-cts` and `aes128-cts` encryption type keys are generated and saved to the keytab file. However, if `arcfour-hmac-md5` encryption is specified, the MD4Hash of the computer password is generated and saved to the keytab file.

In addition, depending on how your environment is configured, you can choose whether to remove any non-AES encryption keys for service principal names (SPNs) from the computer's keytab file by setting the `adclient.krb5.clean.nonfips.enctypes` parameter in the `centrifydc.conf` configuration file. If you set this parameter to true, `adclient` scans the keytab file and removes any non-AES encryption keys for SPNs during startup. This parameter is false by default.

Restarting the agent after enabling FIPS mode

If you use the ADM group policy template, which does not perform validation checks, or if you manually enable FIPS mode by setting the `fips.mode.enable` parameter in the agent configuration file, the `adclient` process will not start if the domain functional level is below Windows Server 2008.

If you attempt to start `adclient` and the domain functional level is below Windows Server 2008, you will see the following error message:

```
Cannot start adclient in FIPS Mode as machine is joined to domain
with Pre-Windows 2008 Domain Functional Level!
```

To restart the agent, you must disable FIPS mode by setting the `fips.mode.enable` parameter to `false` or the “Use FIPS compliant algorithms for encryption, hashing and signing” group policy to Not configured. After disabling FIPS mode, you can continue working at your current domain functional level in non-FIPS mode by restarting the agent:

```
/usr/share/centrifydc/centrifydc restart
```

If you want to enable FIPS mode, leave the current domain, update your domain functional level, then join a Windows Server 2008, or later, domain.

Importing sudoers configuration files

If you are currently managing privileges on Linux and UNIX computers using multiple `sudoers` configuration files, you can import that information and convert it into rights and role definitions that can then be assigned to Active Directory users and groups, local users and groups, or both.

This chapter describes how to migrate all of your privilege management information from `sudoers` configuration files to Active Directory through Access Manager.

Identify the sudoers file on each computer

Most organizations use `sudoers` configuration files and the `sudo` program to manage privileges on Linux and UNIX computers. To read the `sudoers` file on each Linux or UNIX computer, you must have root-level permission. You can define a command right to grant this level of access to other users.

The default location for the `sudoers` configuration file is `/etc/sudoers`, and in general, this is the file to import from each computer. However, there are some exceptions:

- If `sudo` was compiled with the `--sysconfdir` option to specify a different location for `sudoers` file, you need to find the actual location. Run `sudo -V` to see the `sudo` configuration options, including the path to the `sudoers` file.
- If your environment has an automatic mechanism for distributing a single `sudoers` file to the entire network, you can use that one file and don't need to import multiple files.

Get the sudoers file from each computer

You can manually copy the sudoers file to a location on the Windows computer that has Access Manager installed. You should specify a file name that identifies the computer where the file was used. For example, you might include the local host name or a functional description, such as `oracle_server_sudoers.txt` or `qa1_server_sudoers.txt`.

Import the sudoers file

After you have copied the sudoers file to the computer where Access Manager is installed, you can import the sudoers file into a selected Centrify zone.

To import the sudoers file

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you want to import the sudoers file.

   In most cases, if you have a sudoers file that covers multiple computers, you should import it into a parent zone so that it is available to multiple child zones. If the file is used on a single computer, you might select the specific child zone that contains that computer.
3. Right-click, then select **Import sudoers file.**
4. Click **Browse** and navigate to the location in which you copied the sudoers file, select the file, click **Open**, then click **Next**.
5. Review the contents of the file to verify you are ready to import, then click **Next**.

   If you have previously imported a sudoers file—for example, from a different computer—importing a new sudoers file overwrites the data from the previous import. If you have not yet converted the previous sudoers information to rights and rights in Access Manager, click **Cancel** to exit the wizard.

   For more information about convert the imported information to Centrify rights and roles, see Converting sudoers aliases and user specifications before importing an another sudoers file.

6. Review the parsing summary for errors or warnings to verify whether you are ready to import, then click **Next**.

   You can click **Details** to see the list of error and warnings, if applicable. From the list, you can select a specific error or warning, then click **Go To** to see the definition in the `sudoers` file. You can continue with the import if the list only displays warnings. If there are errors, you must fix them before continuing. Make note of any errors and warnings to fix, then click **Close** to close the Details list.

   If the file contains errors, or if you want to fix warnings before importing, click **Cancel** to exit the wizard. You can then open the `sudoers` file in a text editor to fix, delete, or comment out the lines in the file, then save it. After you have modified the file, you can rerun the Import Sudoers File wizard.

7. Click **Finish** to complete the import.

The import wizard creates a new node called Sudoers, which contains sub-nodes for the types of data contained in a `sudoers` file. For example, expand **Sudoers** to see the nodes for User Alias, Runas Alias, Host Alias, Command Alias, and User Specifications. If the Sudoers node is not visible, select Authorization, right-click, then click **Refresh**.

Some or all of the Sudoers sub-nodes might be empty depending on whether the `sudoers` file included definitions of that type. For example, if there are no user aliases defined in the `sudoers` file, the User Alias sub-node is displayed in Access Manager, but there are no entries under it.

You can now convert the `sudoers` data to rights, role definitions, and role assignments in the Centrify zone. If you intend to import more than one `sudoers` file into the same zone, you must convert the imported aliases and user specifications to rights, role definitions, and role assignments before importing another `sudoers` file.

Converting sudoers aliases and user specifications

Before you convert the `sudoers` file aliases and user specifications to rights, role definitions, and role assignments, be certain that you have imported all the users and groups specified in the `sudoers` file into Active Directory, and that you have added them to the zone in which you are imported the `sudoers` file. If there are users and groups without a profile in the zone when you attempt to convert the user specifications from the imported `sudoers` file into role assignments in Access Manager, the conversion will fail.

In addition, keep in mind that the role definitions and assignments you create from `sudoers` specifications do not contain any UNIX system rights or PAM access rights. You can assign those rights through other roles, such as the predefined `UNIX Login` role, or you can add system rights and PAM access rights to the role definitions after you create them from the `sudoers` specifications.

Within each item are objects for the `sudoers` definitions that were imported. For example, within User Alias are alias definitions, each one of which contains the user accounts defined for that alias.

Each type of information from the `sudoers` file converts to a different type of authorization information in the Centrify zone. You do not need to convert all of the imported aliases. You can simply ignore or delete aliases that are obsolete or no longer relevant.

Converting user aliases

On Linux and UNIX computers, a user alias in the `sudoers` file defines a set of users without creating a group. When you convert a user alias specification to be used in a zone, however, it becomes an Active Directory group. Assigning users to groups simplifies user management because if users change roles or leave the company, you can simply remove their group membership, without deleting their accounts, and effectively, they no longer have access to the roles assigned to members of the group.

You can create a new Active Directory group from the user alias you imported or map the imported alias to an existing Active Directory group.

To create a new Active Directory group from a user alias

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the `sudoers` file.
3. Expand Authorization and Sudoers, then select **User Alias**.
4. Select the alias name, right-click, then select **Create AD Group**.

5. Verify the container location, or click **Browse** to select a different container, then click **Next**.
6. Verify the group name, which defaults to the alias name, optionally, add a prefix or suffix, and select the scope for the group, then click **Next**.
7. Review the group and group membership information displayed, then click **Next**.

   If there are any warnings or errors displayed, you must fix the errors before continuing. If only warning are displayed, you can continue to create the group. For example, if the user alias has members that don't have a corresponding Active Directory account, you can continue creating the group.
8. Review information about the new Active Directory group, then click **Finish** to create the group.

To map a user alias to an existing group

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the `sudoers` file.
3. Expand Authorization and Sudoers, then select **User Alias**.
4. Select the alias name, right-click, and select **Map to AD Group**.
5. Select **Remove original AD group membership** or cancel the selection depending on whether you want to keep the current members of the group when adding the users from the alias definition.

   If you select this option, the wizard removes the existing members of the group when adding the new members. If you do not select this option, the wizard adds the new members to the existing members.
6. Click **Browse**, then enter search criteria to identify the group and click **Find Now**.
7. Select the name of the group and click **OK**.

   The wizard imports the users defined by the alias into the specified Active Directory group. It also issues a warning message that it can't import users who are defined by the alias but who are not defined in Active Directory.

Viewing run-as aliases

A run-as alias defines a group of one or more users who other users are able to run commands as. Select and double-click the alias name to expand it and see the users who are defined for it. You cannot directly import run-as aliases. However, if a user specification includes a run-as alias, you can view the run-as definition in the **Runas Alias** node, and import the commands defined in the specification. For more information about user specifications, see Converting user specifications .

Converting host aliases

Host alias definitions are popular in centralized `sudoers` files because they allow you to assign privileges to groups of computers rather than managing privileges on an individual computer and file basis. They convert naturally to computer roles, which also assign privileges to groups of computers.

When you convert a host alias to a computer role, the wizard creates a new computer role, creates an Active Directory group that contains the computers defined in the host alias, and adds these computers to the new computer role. Because the computer role group is an Active Directory group, the computers can span multiple zones and include computers that are joined to different zones. To complete the computer role definition, you must add the appropriate user role assignments, which specify what specific users and groups in different role definitions are allowed to do on the computers included in the computer role group.

To create a computer role from a host alias

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the `sudoers` file.
3. Expand Authorization and Sudoers, then select **Host Alias**.
4. Select the alias name, right-click, then select **Create Computer Role**.
5. Click **Next** to accept the location for the group of computers, or change the location, then click **Next**.

6. Verify or change the group name, optionally, add a prefix or suffix, and select the scope for the group, then click **Next**.
7. Review the group and group membership information displayed, then click **Next**.
8. Review information about the new Active Directory group for computers, then click **Finish** to create the group and the new computer role.

   If the computer accounts exist in Active Directory, the computers defined in the host alias are automatically added to the new Active Directory computer group and to the "Members" node of the new computer role.
9. Expand Authorization, Computer Roles, and the computer role name.
10. Select Role Assignments, right-click, and click **Assign Role**.
11. Select the role and click **OK**.
12. Click **Add AD Account**.
13. Select User or Group, enter search criteria, then click **Find Now** to search for and elect the user or group, then click **OK**.
14. Select the appropriate user or group from the result, then click **OK** to complete the user role assignment.

Viewing command aliases

You can select the Command Alias sub-node to view the command aliases that were imported from the `sudoers` file. You can't edit or delete the command aliases. The information is displayed for your reference. You can assign the command aliases listed role definitions, role groups, and computer roles when you convert the user specifications imported from the `sudoers` file.

Converting user specifications

In the `sudoers` file, user specifications make use of the alias definitions to assign commands and privileges to users. After you import the `sudoers` file, you can convert the user specifications into role assignments.

To convert user specifications to role definitions and role assignments

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the `sudoers` file.
3. Expand Authorization and Sudoers, then select **User Specifications**.
4. Select the name of a user specification, right-click, then select **Import**.
5. Review the list of commands to be created, then click **Next**.
6. Verify the name of the role definition name to be created, then click **Next**.

   By default, the role definition is named `Role`_n. You can change it after it is created.
7. If the user or group defined in the imported user specification is not found in the zone, the role assignment to be created is displayed and you can click **Next**, then click **Finish**.

   If the user or group defined in the imported user specification is not found in the zone, the role assignment will fail and the role displays an error (❌). Click **Cancel** to exit the wizard and add the user or group to Active Directory and the zone.

   Importing a user specification will fail if the user or group defined in the user specification is not found in the zone or if no computers are defined for the host alias in the user specification are found in the zone.
8. Rename the role definition by expanding Authorization and Role Definitions.
   - Select the new role definition, for example, Role_2.
   - Right-click, then select **Rename**
   - Type a new name for the role definition.

The role definitions you create from a `sudoers` specification do not contain the UNIX system rights or PAM access rights. You can assign these rights through a separate role assignment or by add the appropriate UNIX system rights and PAM access rights to the new role definitions.

Removing imported sudoers information

Once you have validated the conversion of imported `sudoers` file information, you can purge the `sudoers` information from Access Manager.

To purge sudoers information

1. Open Access Manager.
2. Expand Zones and the individual parent or child zones required to select the zone name into which you imported the `sudoers` file.
3. Expand Authorization, then select **Sudoers**.
4. Right-click, then select **Purge**.

   The Sudoers node and sub nodes are removed from Access Manager.

Mapping sudo to dzdo

To execute privileged commands users must type `dzdo` and the command name. If you want, you can map `sudo` to `dzdo`, which allows your users, who are accustomed to using `sudo` to execute their privileged commands, to continue to type `sudo` commandName. If the user has a role assignment that allow him to execute the command in an unrestricted shell, the command is executed using `dzdo` commandName. To map sudo to dzdo on computers in your organization, you can enable the "**Replace sudo by dzdo"** group policy for a site, domain, or organizational unit.

For more information about working with group policies, see the *Group Policy Guide*.

Using Centrify OpenLDAP proxy service

This chapter describes the Centrify OpenLDAP proxy service (`centrifydc-ldapproxy`) that you can use to map Active Directory users to UNIX identities to enable access to the files stored on legacy network appliance servers and storage devices. Centrify OpenLDAP also enables Linux and UNIX computers to search Active Directory domain controllers and global catalog servers for any information stored in Active Directory. If you have the appropriate permissions, you can also use the Centrify OpenLDAP proxy service to add, modify, or delete information stored in Active Directory.

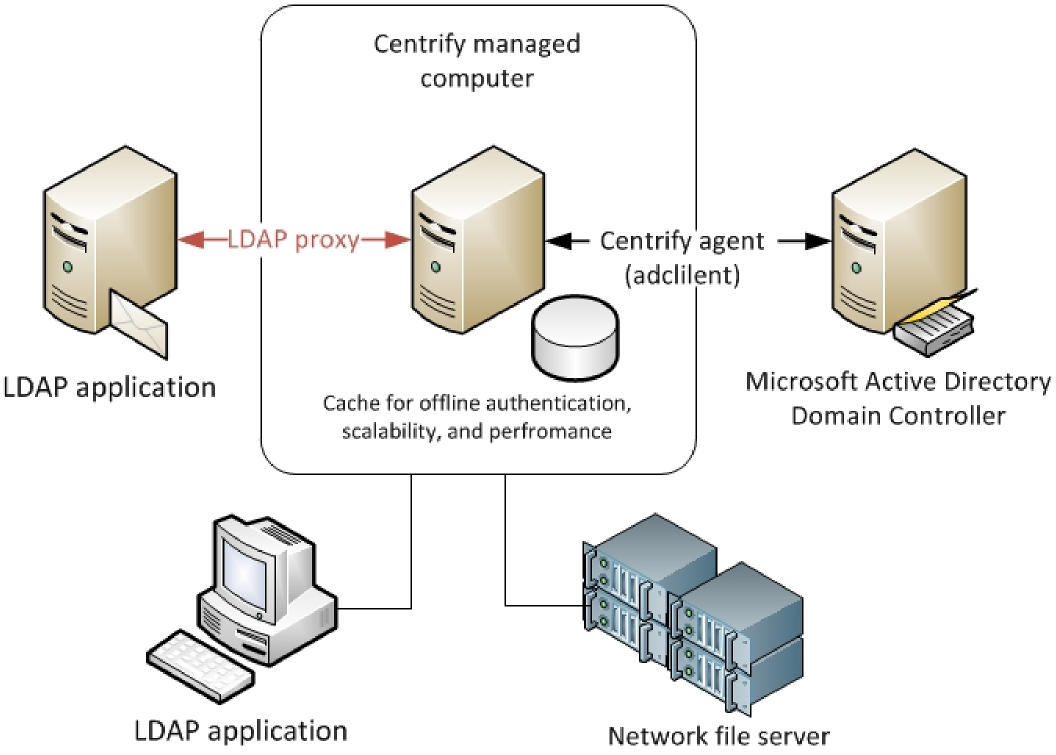
What the OpenLDAP proxy provides

Many applications support the Lightweight Directory Access Protocol (LDAP) and require data stored in this format, but do not support Kerberos. In addition, many applications that support LDAP cannot search Active Directory directly because of the complexities of the Active Directory environment itself, such as the global catalog, multiple domains, multiple forests, and trust relationships.

The Centrify OpenLDAP proxy is an OpenLDAP server process that enables LDAP clients that are not Kerberos-enabled to search Active Directory efficiently and securely. By using the Centrify OpenLDAP proxy, applications that support LDAP can search complex Active Directory environments and authenticate users with Active Directory. Through the Centrify agent, the Centrify OpenLDAP proxy enables you to resolve UID, GID, and group membership efficiently and collapse the entire Centrify hierarchical zone structure, including parent and child zone, and individual computer overrides into a single namespace for LDAP applications.

In addition, connecting to Active Directory typically requires an authenticated bind with a valid user name and password. Because the Centrify OpenLDAP proxy uses the Centrify agent to connect to Active Directory and retrieve information, you can issue OpenLDAP commands without an authenticated bind.

The following diagram provides a simplified overview of the components.


Centrify managed computer
LDAP proxy
Centrify agent (adclilent)
LDAP application
Cache for offline authentication, scalability, and perfromance
Microsoft Active Directory Domain Controller
LDAP application
Network file server

The key advantages to deploying the Centrify OpenLDAP proxy when you have LDAP clients where the Centrify agent cannot be installed are as follows:

- You can use the Centrify OpenLDAP proxy server to run commands that retrieve or update information stored in Active Directory.
- The Centrify OpenLDAP proxy service uses the Centrify agent to securely connect to Active Directory and retrieve user, group, and other information from the Active Directory domain controller.
- You can leverage the offline authentication and caching capabilities of the Centrify agent for applications that support LDAP, but not Kerberos.
- Regardless of the complexity in Active Directory, including multiple domains and forests and parent and child zones, the Centrify OpenLDAP proxy treats the information stored in Active Directory as a single RFC2307-compatible namespace.

• • • • • •

Enabling simple authentication

Users can be authenticated through simple authentication to the Centrify OpenLDAP proxy with their username and password. This is then converted to a secure Kerberos authentication by adclient.

By default, to authenticate users, adclient checks its credential cache data first, then, if not in cache, it refers to Active Directory. Allowing Centrify OpenLDAP proxy to use the adclient credential cache, enables authentication if adclient is in disconnected mode.

If you want to always authenticate through Active Directory:

To the `slapd.conf` file:

```
/etc/centrifydc/openldap/slapd.conf
```

Add:

```
cdc-auth-prefer-cache false
```

Enabling simple proxy mode

If either `objectClass` or `objectCategory` is not specified in the search filter, the search is in simple proxy mode. With simple proxy mode, all search filters are sent without translation through adclient to Active Directory. All results are returned as provided by Active Directory without translation or interpretation of results.

Accessing network appliance or storage servers

One of the most common uses for the Centrify OpenLDAP proxy service is to provide access to the files stored on legacy network appliance file servers and storage devices. Many organizations use network appliance file servers and storage area network devices to provide highly available and scalable data storage services that support multiple client access protocols—including NFS, CIFS and iSCSI—and multiple operating systems.

Supporting multiple protocols and operating systems, however, presents a challenge when users want to access files from computers with different operating systems. To ensure users are granted proper access to files stored on a network appliance or storage server, their identity attributes must be consistently defined for both UNIX and Windows operating systems.

For example, the identity attributes that allow access to the files on a network appliance or storage server might be UNIX profile attributes from a common NIS or LDAP repository. The UID and GID values establish file ownership and file access permissions. For Windows users to access the files stored on the network appliance or storage server, their Windows account must be mapped to the UNIX profile that grants them the appropriate file permissions.

Mapping Active Directory users to UNIX profiles

Centrify enables you to map Active Directory users to one or more UNIX profiles. The UNIX profile contains each user's identity attributes. You can use this mapping of Active Directory account information to UNIX identity attributes to provide consistent file and directory ownership and access rights to files that are stored on a network appliance or storage server.

By mapping an Active Directory account to a UNIX profile, you can ensure that a user's identity is consistently maintained and that access to UNIX-hosted resources is properly protected regardless of the computer from which the user accesses the resource.

For network appliance or storage servers that are hosted on UNIX computers and require UNIX identity attributes to grant access, you can use the Centrify OpenLDAP proxy service to make the Active Directory-hosted user mapping information available through the LDAP or LDAPS protocol.

Configuring servers to use the proxy service

Before you can use the Centrify OpenLDAP proxy service to look up information stored in Active Directory, the network appliance, storage device, or file server you want to use must be configured to use LDAP to look up user and group information. In most cases, this is an option you configure when setting up a server or device.

If your vendor supports connecting to LDAP servers for authentication and authorization services, configuring the server or device to use the Centrify OpenLDAP proxy requires the following high-level steps:

1. Install Access Manager, create at least one zone, and add users to the zone.
2. Install the Centrify agent on a Linux or UNIX computer and join the computer to an Active Directory domain.
3. Install the `centrifydc-ldapproxy` package on the Linux or UNIX computer.
4. Start the `centrify-ldapproxy` service and verify proper operation.
5. Set up the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service to look up user and group information.
6. Test the solution for proper end-to-end operation.

Installing the Centrify OpenLDAP proxy service

On most platforms, the `centrifydc-ldapproxy` package is available with the Centrify agent software package but is not installed by default. You can select the package in the installation script or install it using a native package installer.

To run the Centrify OpenLDAP proxy service, the computer must:

- Be joined to an Active Directory domain.
- Have the Centrify agent installed and the adclient running.

In the following example, the agent is installed on a Linux computer and the computer is joined to the `pistolas.org` Active Directory domain.

To install the Centrify OpenLDAP proxy service on a Linux computer

1. Log on or switch to the `root` user, then navigate to the directory where you extracted Centrify files.

   For example, if you ran the `gunzip` and `tar` commands in the `/tmp` directory, change to the `/tmp` directory.

2. Run `install.sh` or a native package manager to install the files.

   For example, run the following command:

   ```
   ./install.sh
   ```

You can type `K` to keep any existing packages you have installed. When you see the `Install the CentrifyDC-ldapproxy package` prompt, type `Y`. Follow the remaining prompts displayed to complete the installation.

Alternatively, you can use a native package manager. For example on most Linux distributions, you can run a command similar to this:

```
rpm -Uvh centrifydc-ldapproxy-release-arch.rpm
```

If you are installing on Solaris, unzip and extract the contents of the package, then run a command like this:

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

If you are using an installation program, such as SMIT or YAST, see the documentation for that program.

3. If you want to start the `ldapproxy` service with parameters, configure the `STARTUP-OPTS` option.

   Run the appropriate command for your platform.

   - For CentOS, SLES

     ```
     echo "STARTUP_OPTS=\"-h ldaps:///\"" >>
     /etc/sysconfig/centrify-ldapproxy
     ```

   - For Debian

     ```
     echo "STARTUP_OPTS=\"-h ldaps:///\"" >>
     /etc/default/centrifyldapproxy
     ```

   - For HPUX

     ```
     echo "STARTUP_OPTS=\"-h ldaps:///\"" >>
     /etc/rc.config.d/centrify-ldapproxy
     ```

   - For AIX

     ```
     chssys -a "-d 0 -h ldaps:///" -s centrify-ldapproxy
     ```

   - For Solaris without Service Management Facility (SMF)

     ```
     echo "STARTUP_OPTS=\"-h ldaps:///\"" >>
     /etc/centrifydc/openldap/centrify-ldapproxy.conf
     ```

   - For Solaris with Service Management Facility (SMF)

     ```
     svccfg -s centrify-ldapproxy setprop 'slapd/STARTUP_
     OPTS=("-h""ldaps:///")'
     ```

4. Start the `centrify-ldapproxy` service.

   For example, on Linux computers:

   ```
   /usr/share/centrifydc/bin/centrify-ldapproxy start
   ```

5. Test the service by searching for an object in the Active Directory domain.

For example, to search for groups in the domain, you might type commands like this:

```
cd /usr/share/centrifydc/bin
ldapsearch -h localhost -p 389 -x -b “dc=pistolas,dc=org”
-s sub "objectClass=group" -D
"cn=amy.adams,cn=users,dc=pistolas,dc=org" -w password
```

The `-h` and `-p` options are required to connect to Active Directory using the proxy service and the Centrify agent. If the LDAP proxy service is not on the local computer, use the `-h` option to specify the name of the computer where you have installed it.

You can also connect to Active Directory directly using a valid user name and password. For example:

```
ldapsearch -D "cn=amy.adams,cn=users,dc=pistolas,dc=org" -W
-h dc2012.pistolas.org -p 389 -x -b "dc=pistolas,dc=org"
-s sub "objectClass=group"
```

6. (Optional) Review and modify, if necessary, the default `centrify-ldapproxy` service start-up script in the `/etc/init.d/` directory.

   You can use the `/usr/share/centrifydc/bin/centrify-ldapproxy` script to start, stop, restart or check the status of the Centrify OpenLDAP proxy service.

   Note: By default, the service starts automatically when the computer restarts.

Specifying the LDAP server

After you have installed and tested the Centrify OpenLDAP proxy service, the next step is to configure the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service to look up user and group information. In most cases, this involves setting configuration options to specify the computer where the Centrify OpenLDAP proxy service is running as the LDAP server you want to use in a local or system-wide `ldap.conf` file. You should consult the documentation provided by the vendor you are integrating with for details about how to set up LDAP integration.

Testing the solution

After you have configured the network appliance, storage device, or file server to use the Centrify OpenLDAP proxy service on a Centrify-managed computer, you should verify that files created by a Windows user have the correct UID and GID to access those files from both a UNIX computer and a Windows computer.

Manually starting the OpenLDAP service

Typically, the Centrify OpenLDAP proxy service automatically starts when the computer restarts. You have the option to manually start Centrify OpenLDAP proxy service.

The Centrify OpenLDAP proxy service is a modified version of the standard `slapd` LDAP server process. The Centrify version of the slapd process also uses a customized version of the standard LDAP server configuration file in the following location:

```
/etc/centrifydc/openldap/slapd.conf
```

This customized version of the `slapd.conf` configuration file is created automatically when you join a domain and is configured by default with Access Manager and domain-specific information. You can start or stop the `slapd` process at any time by using the `centrify-ldapproxy` script or directly from the command line.

To start the Centrify OpenLDAP proxy service directly from the command line using the default configuration file, you can run the following command:

```
/usr/share/centrifydc/libexec/slapd
```

or

```
/usr/share/centrifydc/bin/centrify-ldapproxy start
```

If you start the `slapd` process directly from the command line, you can also specify additional command line options just as you would for the standard `slapd` LDAP server process. For example, you can use the `-f` command line option to specify a different configuration file to use:

```
slapd -f /etc/centrifydc/openldap/slapd.conf
```

or

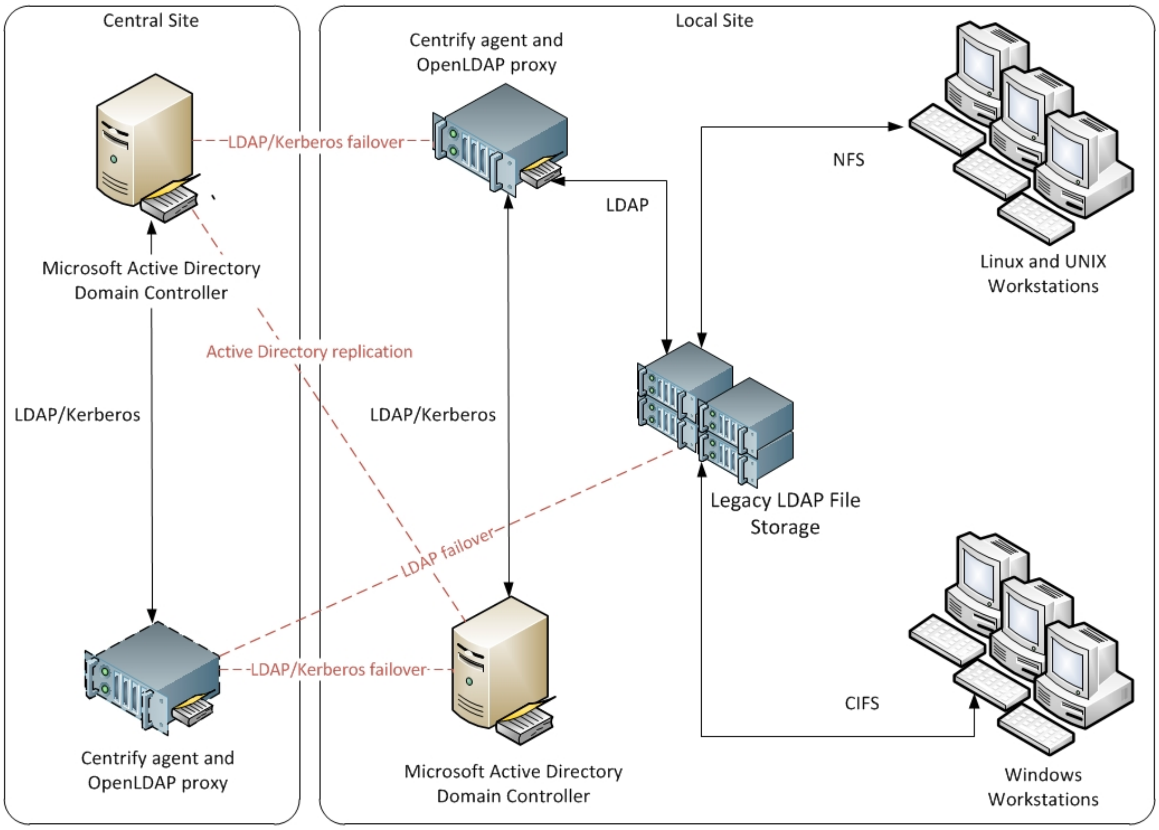
```
/usr/share/centrifydc/bin/centrify-ldapproxy start -f /
 etc/centrifydc/openldap/slapd.conf
```

For more information about the command line options when starting the LDAP server directly from the command line, see the `man` page for the `slapd` process.

> **Note:** On the computer that is using systemctl, if the slapd process crashes, the systemd process will restart the slapd process.

Sample deployment scenario

The following diagram illustrates a basic deployment scenario for a distributed site with minimal OpenLDAP proxy overhead. As depicted in this illustration, the legacy LDAP servers and devices support a redundant LDAP server for fault tolerance and failover.


Central Site
Local Site
Centrify agent and OpenLDAP proxy
LDAP/Kerberos failover
Microsoft Active Directory Domain Controller
LDAP
NFS
Linux and UNIX Workstations
Active Directory replication
LDAP/Kerberos
LDAP/Kerberos
Legacy LDAP File Storage
LDAP failover
LDAP/Kerberos failover
CIFS
Centrify agent and OpenLDAP proxy
Microsoft Active Directory Domain Controller
Windows Workstations

Using OpenLDAP commands

The Centrify OpenLDAP proxy service includes a set of OpenLDAP commands that have been modified to support looking up information in Active Directory domain controllers and the global catalog. The Centrify distribution of OpenLDAP supports most of the standard options and syntax for performing LDAP operations, but the `ldap` commands in the Centrify distribution of

OpenLDAP also support the following options that are not supported in a standard OpenLDAP distribution:

Use this option	To do this
`-m`	Use the local machine credentials from the `/etc/krb5.keytab` file. This option requires `root` user access.
`-r`	Disable line wrapping when printing out LDIF entries.

The Centrify distribution of OpenLDAP also provides extended URL support for Active Directory. With Centrify LDAP commands, you can use the following URLs to connect to Active Directory computers:

Use this	To do this
`ldap://domain_name`	Connect to the appropriate domain controller for the specified domain within the Active Directory site.
`ldap://`	Connect to the joined domain.
`gc://[domain_name]`	Connect to the global catalog domain controller for the joined domain. You can use the optional domain_name parameter to specify a domain in a different forest.

The Centrify distribution of OpenLDAP includes the following commands:

- ldapsearch
- ldapadd
- ldapmodify
- ldapmodrdn
- ldapcompare
- ldapdelete

Note: The `ldappasswd` and `ldapwhoami` commands do not work with Active Directory. For more information about using the OpenLDAP commands or the standard options available, see the `man` page for each command.

Centrify OpenLDAP proxy commands attributes

The Centrify OpenLDAP proxy commands accept the following attributes.

- dn - Specifying the dn attribute returns only the distinguished name
- `1.1` - Specifying the `1.1` attribute returns only the distinguished name
- `*` - Specifying the asterisk (*) attribute return is situational:
    - If only `*` is specified, Centrify OpenLDAP proxy returns all our supported attributes.
    - If the `*` is specified with additional attributes, Centrify OpenLDAP proxy returns the given additional attributes.

Searching for users and groups

If you want to use `ldapsearch` to find a user, do not use `objectclass=user` or `objectcategory=person` to specify the filter. Instead, you should use `objectclass=posixaccount`. For example, to find the user with the UNIX name `jtr` enter a command similar to the following:

```
/usr/share/centrifydc/bin/ldapsearch -x -h localhost -D
“CN=Administrator,CN=Users,DC=pistolas,DC=org” -W -b
“dc=pistolas,dc=org” "(&(objectclass=posixaccount)(uid=jtr))"“
```

Optionally, use the UID number instead of the UNIX name:

```
"(&(objectclass=posixaccount)(uidNumber=1234567))"
```

Similarly, use `objectclass=posixgroup` to retrieve information on a group. This filter supports the following options:

- `cn`: Find a group with a given UNIX name
- `gidNumber`: Find a group with a given GID
- `memberUID`: Search for secondary group membership of given UNIX user.

Searching the global catalogs

In most cases, you use the Centrify OpenLDAP proxy service to search for information through the domain controller. However, you can also use the Centrify OpenLDAP proxy service to perform searches in the global catalog, if needed. The global catalog search is especially useful if you have a large, multiple-domain forest.

To specify that you want the Centrify OpenLDAP proxy service to search the global catalog, add “`CN=$`” to the front of the search base.

To search Active Directory for a specific account, use the syntax:

```
"(&(objectCategory=Person)(Name=amy.adams*))"
```

For example, in the global catalog, you might type a command similar

to the following:

```
/usr/share/centrifydc/bin/ldapsearch -h localhost -D
"cn=amy.adams,cn=NewUsers,dc=ajax,dc=org" -w password -x -b
"cn=$"
```

By default the Centrify OpenLDAP proxy service is configured to disable anonymous binds. To allow anonymous binds:

1. Edit the `/etc/centrifydc/openldap/slapd.conf` file.
2. Remove or comment following line.

   ```
   require authc
   ```

If anonymous binds are disabled, you no longer need to specify the `-D` and `-w` parameters to invoke an `ldapsearch`. For example:

```
ldapsearch -h localhost -x -b "dc=wonder,dc=land"
"(&(objectClass=User)(displayName=Mister\*))" displayName
```

Minimizing search traffic to adclient

To minimize the traffic to adclient and subsequently to Active Directory, during an `ldapsearch`, the Centrify OpenLDAP proxy implements memory cache. The Centrify OpenLDAP proxy memory cache is disabled by default.

To enable the Centrify OpenLDAP proxy memory cache, change `slapd.conf` to:

```
ldapproxy.cache.enabled true
```

Enabling encrypted communication

By default, communication between LDAP clients and the Centrify OpenLDAP proxy service is not encrypted. To secure communications between LDAP clients and the Centrify OpenLDAP proxy service using Transport Layer Security (TLS), you must create or obtain the required certificates and configure both the LDAP client and the LDAP server to use the certificates. In addition, you must configure the LDAP server with the certification authority (CA) certificate, its own server certificate, and a private key.

The current versions of the `ldapsearch` client and `ldapproxy` server support Transport Layer Security (TLS) v1.2.

Depending on your network topology, you might also need to modify client-side or server-side configuration settings to successfully return search results.

Preparing for auto-enrollment

You can configure the Centrify OpenLDAP proxy service to automatically get the certificate, private key, and CA chain for secure LDAP (`ldaps`) connections. To configure automatic enrollment for certificates, however, you must have an Active Directory domain controller that you can use as a certification authority for issuing certificates.

The following steps summarize how to prepare the domain controller:

1. Use Server Manager to add the Active Directory Certificate Services role to a domain controller.
2. In the Add Roles wizard, select the Certification Authority role service and follow the prompts displayed to configure the server role.
3. Open the Certificates MMC snap-in, select the domain controller certificate, right-click, then click Open.
4. Select the Details tab, click Copy to file, then follow the prompts displayed to export the certificate to a file.
5. From Administrative Tools, select Group Policy Management, then select an appropriate Group Policy Object for the forest and domain you want to edit.
6. Right-click the Group Policy Object, then click **Edit**.
7. Under Computer Configuration, expand Policies > Windows Settings > Security Settings, then select Public Key Policies.
8. Select Trusted Root Certificate Authorities, right-click to select **Import**, then follow the prompts displayed to import the certificate.
9. Select Certificate Services Client - Auto-Enrollment, then select **Enabled**.
10. From Administrative Tools, select Certification Authority, expand the name of the domain controller you are using as the certification authority, then select **Certificate Templates**.

11. Right-click to select Manage, select an appropriate template to use, such as the Computer template, right-click, then click **Duplicate Template** to open the properties page for the new template.

12. Type an appropriate name for the new template, such as Centrify OpenLDAP Proxy.

13. Click the Security tab, select the Domain Computers group, select Allow for the Autoenroll permission, then click **Apply**.

    You can set other properties on the remaining tabs, as needed. For example, you might want to click the Subject Name tab to change the subject name format to Fully distinguished name. When you are finished setting properties for the template, click **OK**.

14. In the Certification Authority console, select Certificate Templates, right-click to select New, then click **Certificate Template to Issue**.

15. Select the template you created, for example, select the Centrify OpenLDAP Proxy template, then click **OK**.

Updating the Centrify OpenLDAP proxy computer

After you have prepared the domain controller with the policy for certificate auto-enrollment, you can use the following steps to provide the required certificate, private key, and certification authority.

1. Verify the computer where you are running the Centrify LDAP proxy service is joined to an Active Directory domain.

2. Change to the directory where certificates for auto-enrollment are located.

   ```
   cd /var/centrify/net/certs/
   ```

   You should see files similar to the following listed in the directory:

   ```
   auto_LDAPProxy.cert
   auto_LDAPProxy.chain
   auto_LDAPProxy.key
   trust_41DFF689876FCE52E02EE73FC7E3782964DC54BB.crl
   trust_F7842B2A65489F15A1722518E41F5E6B0F4FBC5E.cert
   ```

3. Run an `openssl` command similar to the following to create the certificate:

   ```
   openssl pkcs7 -in auto_LDAPProxy.chain -text -out auto_
   LDAPProxy_CA.pem -print_certs
   ```

4. Add the following lines to `/etc/centrifydc/openldap/slapd.conf`

configuration file. Comment out the old TLSCipherSuite line, as shown here.

```
TLSCertificateFile /var/centrify/net/certs/auto_LDAPProxy_
CA.pem
TLSCertificateFile /var/centrify/net/certs/auto_
LDAPProxy.cert
TLSCertificateKeyFile /var/centrify/net/certs/auto_
LDAPProxy.key
TLSCipherSuite TLSv1.2
# TLSCipherSuite SSLv3
```

You should also review and modify other server configuration settings, if needed. For example, you might use settings similar to the following:

```
# Require START TLS on port 389
security tls=1
# Require TLS v1.0 or better
TLSProtocolMin 3.1
TLSVerifyClient try
```

5. Add the following line to /etc/centrifydc/openldap/ldap.conf configuration file:

   ```
   TLS_CACERT /var/centrify/net/certs/auto_LDAPProxy_CA.pem
   ```

   You should also review and modify other configuration settings, if needed. For example, you might need to change the TIMEOUT value to allow clients to wait an appropriate number of seconds for a response:

   ```
   TIMEOUT 15
   ```

6. Restart the Centrify OpenLDAP proxy service.

   ```
   sudo /usr/share/centrifydc/bin/centrify-ldapproxy start -h
   ldaps:///
   ```

7. Test operation by running an OpenLDAP command, such as ldapsearch.

   ```
   /usr/share/centrifydc/bin/ldapsearch -x -H
   ldaps://localhost:636 -b 'cn=users,dc=win2012,dc=test' -D
   administrator@win2012.test -W "(cn=test_user)"
   ```

8. To confirm that TLSv1.2 is being used, use openssl s_client to connect to the slapd. For example, enter:

   ```
   $ openssl s_client -connect localhost:636 -showcerts -state -
   CAfile /etc/centrifydc/openldap/cacert.pem
   ```

9. Review the output from the previous command and confirm that the protocol is TLSv1.2, as shown here:

   ```
   ...
   SSL Session:
           Protocol : TLSv1.2
   ```

10. (Optional) Alternatively, to confirm that TLSv1.2 is used, run a software tool like Wireshark to capture and inspect the `ldapsearch` traffic.

Securing communication without auto-enrollment

If you are not using an Active Directory domain controller and auto-enrollment for certificate distribution, you can manually configure the Centrify OpenLDAP proxy service to use the server certificate and private key you create.

The following steps summarize how you can manually configure the Centrify OpenLDAP proxy service to use certificates.

1. Use CA.sh to create the certificates:

   ```
   /usr/share/centrifydc/ssl/misc/CA.sh -newca
   ```

   ```
   /usr/share/centrifydc/bin/openssl req -new -nodes -keyout newreq.pem
   -out newreq.pem
   ```

   ```
   /usr/share/centrifydc/ssl/misc/CA.sh -sign
   ```

2. Install the certificates in the `/etc/centrfydc/openldap` directory.

   ```
   cp demoCA/cacert.pem /etc/centrifydc/openldap/cacert.pem
   ```

   ```
   mv newcert.pem /etc/centrifydc/openldap/servercrt.pem
   ```

   ```
   mv newreq.pem /etc/centrifydc/openldap/serverkey.pem
   ```

3. Add the following lines to `/etc/centrifydc/openldap/slapd.conf` configuration file:

   ```
   TLSCACertificateFile /etc/centrifydc/openldap/cacert.pem
   ```

   ```
   TLSCertificateFile /etc/centrifydc/openldap/servercrt.pem
   ```

   ```
   TLSCertificateKeyFile /etc/centrifydc/openldap/serverkey.pem
   ```

4. Add the following line to `/etc/centrifydc/openldap/ldap.conf` configuration file:

   ```
   TLS_CACERT /etc/centrifydc/openldap/cacert.pem
   ```

5. Start the slapd deamon using the following:

   ```
   /usr/share/centrifydc/libexec/slapd -h "ldaps:///"
   ```

   or

   ```
   sudo /usr/share/centrifydc/bin/centrify-ldapproxy start -h ldaps:///
   ```

Searching for automount maps and entries

You can use the Centrify `ldapproxy` service and `ldapsearch` command to find `automount` maps and `automount` map entries that you have stored in Active Directory. The following examples illustrate how to write filters to retrieve `automount` information. These examples assume you have added the following `automount` maps and map entires to Active Directory using Access Manager:

- The `auto.home` map has the map entries `test1` and `test2`.
- The `autotest` map has the map entries `test10` and `test11`.

To retrieve both maps, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test" "
(objectClass=automountMap)"
-D "cn=amy.adams,cn=users,dc=pistolas,dc=org" -w password
```

The command returns attribute information for each map similar to this:

```
dn: cn=auto.home,cn=NisMaps,cn=global,cn=Zones,dc=acme,dc=test
automountMapName: auto.home
ou: auto.home
cn: auto.home
displayName: $CimsAutomountMapVersion1
objectClass: top
objectClass: automountMap
uSNChanged: 20046
```

To retrieve information for a specific map, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test"
"(&(objectClass=automountMap)(automountMapName=auto.home))"
-D "cn=amy.adams,cn=users,dc=pistolas,dc=org" -w password
```

To retrieve all map entries from both maps, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "DC=acme,DC=test" "
(objectClass=automount)"
-D "cn=amy.adams,cn=users,dc=pistolas,dc=org" -w password
```

To retrieve information for a specific map entry, you might run a command with a filter like this:

```
ldapsearch -x -h localhost -b "cn=auto.home,DC=acme,DC=test"
"(&(objectClass=automount)(automountKey=test1))"
-D "cn=amy.adams,cn=users,dc=pistolas,dc=org" -w password
```

For information about adding and managing NIS maps to Active Directory, see the *Network Information Service Administrator's Guide*.

Automatic translation to search for zone users

If you integrate the Centrify agent with a software environment that has limited configuration options, a standard ldapsearch query might fail to return zone users and groups. If you encounter this issue, you can use a configuration parameter to automatically translate a standard search for Active Directory users and groups into a search query for zone users and groups.

You can set the ldapproxy.cdctranslate.fetchbydnuid parameter in the slapd.conf configuration file to true if you want a search for Active Directory users and groups to be automatically translated into a search for zone users and groups. The default is false. After changing the parameter setting, you should restart the centrify-ldapproxy service.

Note that the translation only applies if the ldapproxy.cdctranslate.fetchbydnuid parameter is set to true, and the following additional conditions are in the search request:

- For the search base, the first part of the DN must be "uid=unixname"
- The search scope base must be (0)
- The search filter must be (objectClass=*)

For example, automatic translation is performed if you run a command similar to the following after changing the ldapproxy.cdctranslate.fetchbydnuid parameter to true and restarting the centrify-ldapproxy service:

```
ldapsearch -x -D "cn=zoe,OU=ajax,dc=pistolas,dc=org" -w password
-h localhost "(objectClass=*)" -
b "uid=zoe,OU=ajax,dc=pistolas,dc=org"
-s base
```

Using workstation mode and Auto Zone

For most organizations, adding Linux or UNIX computers to an Active Directory domain involves creating one or more parent zones, adding Active Directory users and groups to the zone, and assigning one or more roles to the zone users. As an alternative to this process, you can create a single Auto Zone for all Active Directory users or a specific subset of Active Directory users.

Profiles are generated for all users in the forest

If you use Auto Zone, all of the profile attributes that are normally defined in the zone to which a computer is joined are generated automatically based on user attributes in Active Directory or based on a set of agent configuration parameters. By default, all Active Directory users and groups in for the forest automatically become valid users and groups on the computers joined to Auto Zone. The generated profiles have a unique UID and GID for each Active Directory user in the forest. The generated profile is what enables access to the computers joined to Auto Zone.

In addition, if you have Active Directory users in another forest that has a two-way trust relationship with the forest of the joined domain, all of those users are also valid users for the joined computer.

Note: Auto Zone does not support one-way trusts. If a computer is joined to a domain that has a one-way trust relationship with another domain, the users and groups in the trusted domain do not become valid users and groups on the computer.

Limiting users and groups in Auto Zone

If you want to use Auto Zone, but do not want to give all Active Directory users and groups a valid profile, you can use group policies or configuration settings to limit the generation of profiles and computer access to specific users and groups. For example, if you have a two Active Directory groups with users who need access to Linux, UNIX, or Mac OS X computers, you can use either group policies or configuration settings to specify that only the two groups who require profiles are valid Auto Zone users and all other Active Directory users should be ignored.

Auto Zone does not provide zone-specific features

If you decide to use Auto Zone, you should keep in mind that Auto Zone does not support any zone-specific features, such as the ability to define rights and roles, assign roles to users and groups, configure auditing, or keep legacy identity attributes on different computers.

If you want to configure role-based access rights, delegate administrative activity, or migrate existing users and groups, you should not use Auto Zone. If you have a large Active Directory forest, but only require automatic profile generation for a subset of users and groups, you might want to use a combination of hierarchical zone and Auto Zone.

Joining a domain as a workstation

Auto Zone is created automatically in Active Directory if you join a domain by running the `adjoin` command with the `--workstation` option.

What to do before joining Auto Zone

Before joining a computer to Auto Zone, be certain that the following are true:

- Active Directory identities are unique for the forest and any two-way trusted forest.

- The Active Directory users and groups require a single set of properties for all computers that join the domain through Auto Zone and do not need to be segregated into zones for any reason.
- All domains in the forest and any trusted external forest must be unique or the join will fail. In this case, you must manually configure a unique prefix for each trusted domain using configuration parameters.

Who should perform this task

A Linux or UNIX administrator with `root` permission on the computers you want to join to an Active Directory domain. The administrator must also know the password for an Active Directory domain administrator account.

How often you should perform this task

In most cases, you only do this once for each Linux or UNIX computer that needs to join an Active Directory domain as a workstation.

Rights required for this task

You must have an account with `root` permission to modify agent configuration files on managed computers or an administrative account with write permission to enable group policies on a Group Policy Object linked to a domain or organizational unit.

Steps for completing this task

The following instructions illustrate how to join Auto Zone using the `adjoin` command.

To join a computer to a domain as a workstation

1. Log on the computer with the Centrify agent using an account with `root` privilege.
2. Open a terminal and execute the following command:

```
adjoin domainName --workstation
```

For example:

```
[root@rhe5]#adjoin acme.com --workstation
```

3. Type the Active Directory administrator’s password.

   ```
   Administrator@ACME.COM’s password:
   ```

   ```
   Using domain controller: win-f7d27u7kl6m.acme.com
   writeable=true
   Join to domain:acme.com, zone: Auto Zone succesful
   ```

4. Run the `adinfo` command to verify the connection to Auto Zone:

   ```
   [root@rhe5]# adinfo
   Local host name:   rhe5
   Joined to domain:  acme.com
   Joined as:         rhe5.acme.com
   Pre-win2K name:    rhe5
   Current DC:        win-f72d7u7kl6m.acme.com
   Preferred site:    Default-First-Site
   Zone:              Auto Zone
   Last password set: 2012-09-30 18:08:34 PDT
   CentrifyDC mode:   connected
   Licensed Features: Enabled
   ```

Generating profiles for specific users and groups

You can automatically generate profiles for specific users and groups by enabling group policies in a Group Policy Object for a domain, site, or organizational unit in an Active Directory forest or by specifying configuration settings on individual computers.

Rights required for this task

You must have an account with `root` permission to modify agent configuration files on managed computers or an administrative account with write permission to enable group policies on a Group Policy Object linked to a domain or organizational unit.

Who should perform this task

A Windows or UNIX administrator performs this task, depending on your organization’s policies. In most cases, a Windows administrator is responsible for configuring group policies and modifying Group Policy Objects. If your organization uses local configuration settings, the UNIX administrator is usually responsible for this task.

Steps for completing this task using group policies

In most cases, you should use group policies in a Group Policy Object to identify the Active Directory users and groups for which you want to automatically generate profiles. The Group Policy Object enables you to centrally manage access to computers in the Auto Zone. You can enable and configure the following group policies to specify a subset of Active Directory users and groups that should have access to computers in Auto Zone:

- Specify AD users allowed in Auto Zone
- Specify groups of AD users allowed in Auto Zone
- Specify AD groups allowed in Auto Zone

The following instructions illustrate how to limit the valid users and groups in the Auto Zone using these group policy settings.

To specify users and groups to include in Auto Zone by using group policy settings

1. Identify or create an Active Directory group that includes all of the users that you want to give access to Centrify-managed computers.

   The group can be a domain local, global, or universal group. The group can include sub groups — members of these sub groups will also be included in Auto Zone.

2. Open Group Policy Management to create or select a Group Policy Object that is linked to a site, domain, or organizational unit.

3. Right-click the Group Policy Object, then select **Edit** to open Group Policy Management Editor.

4. Expand Computer Configuration > Policies > Centrify Settings > DirectControl Settings, click Adclient Settings.

   - Double-click “Specify groups of AD users allowed in Auto Zone” to specify users by Active Directory group without automatically generating profiles for the groups themselves.

- Double-click Specify AD users allowed in Auto Zone to specify individual Active Directory users for which to automatically generate profile.
- Double-click Specify AD groups allowed in Auto Zone to specify individual Active Directory groups for which to automatically generate profile.

5. Select Enabled, then click List to browse for the groups or users to include.
6. Click **Add**, enter search criteria, then click **Find Now**.
7. Select one or more groups or users from the list, then click **OK**.

Steps for completing this task using configuration parameters

In some cases, you might want to limit the Active Directory users and groups who have a profile generated by configuring parameters in the centrifydc.conf file on individual computers. For example, you might want to use configuration parameter settings if you don't want to implement or apply group policies on certain computers.

You can configure the following configuration parameters to specify a subset of Active Directory users and groups that should have access to computers in Auto Zone:

- `auto.schema.allow.users`
- `auto.schema.allow.groups`
- `auto.schema.groups`

The following instructions illustrate how to limit the valid users and groups in the Auto Zone using these configuration parameters settings.

To specify users and groups to add to Auto Zone by using configuration parameters

1. On a Windows computer, in Active Directory Users and Computers, identify or create a group or group that includes all the users who you want to have access to your Centrify-managed computers.
2. On each computer to add to Auto Zone, open the `/etc/centrifydc/centrifydc.conf` configuration file.

- Find the `auto.schema.allow.groups` parameter and remove the comment (#) to add the names of groups separated by commas.
- Find `auto.schema.allow.users` and remove the comment (#) to add the names of users separated by commas.
- Find `auto.schema.groups` and remove the comment (#) to add the names of groups separated by commas.

The configuration file contains comments that list the valid formats for user and group names. For more information about setting these parameters or editing the configuration file, see the*Configuration and Tuning Reference Guide*.

3. Save and close the file.

Troubleshooting authentication and authorization

This chapter describes how to use diagnostic tools and log files to retrieve information about the operation of Centrify software and how to identify and correct problems within your environment.

Diagnostic tools and log files

All Centrify services include diagnostic tools and logging mechanisms to help you trace the source of problems if they occur. These diagnostic tools and log files allow you to periodically check your environment and view information about Centrify operation, your Active Directory connections, and the configuration settings for individual UNIX and Linux computers.

Although logging is not enabled by default for performance reasons, log files provide a detailed record of Centrify agent (`adclient`) activity. This information can be used to analyze the behavior of `adclient` and communication with Active Directory to locate points of failure. However, log files and other diagnostic tools provide an internal view of operation and are primarily intended for Centrify experts and technical staff.

In most cases, you should only enable logging when you need to troubleshoot unexpected behavior, authentication failure, or problems with connecting to Active Directory or when requested to do so by Centrify Support. Other troubleshooting tools, such as command line programs, can be used at any time to collect or display information about your environment.

Analyzing information in Active Directory

One important way you can troubleshoot your environment is by running the Analyze command. The Analyze command enables you to selectively check the integrity of the information stored in Active Directory. With the Analyze wizard, you can check for a variety of potential problems, such as duplicate user IDs, duplicate groups, empty zones, orphaned data objects, or computers that have joined more than one zone.

> **Note:** When you run the Analyze command, only the zones that are open are checked.

To check for problems with information in the Active Directory forest:

1. Open Access Manager.

   If you are prompted to connect to a forest, specify the forest domain or domain controller to which you want to connect.

2. In the console tree, select the Access Manager root node, right-click, then click **Analyze**.
3. Select the types of checks you want to perform, then click **Next** to generate the report.

Select this option	To do this
All	Perform all of the data integrity checks. **Note** If you do not register the administrative notification handler through the Setup Wizard or manually using ADSI, you should periodically run the Analyze command with **All** or **Orphan UNIX data objects** selected.
Computers joined to multiple zones	Check for computers that have joined the domain using more than one zone. Each UNIX computer should only reside in one zone, but if you run the join command more than once, it is possible to have the same computer in more than one zone. This option checks for this problem.
Cyclic zone hierarchy	Check for a circular zone hierarchy. The console prevents you from creating a circular zone hierarchy, but it is possible to do so inadvertently when using ADEdit.

Select this option	To do this
Duplicate groups in zones	Check for duplicate UNIX group names or group identifiers (GIDs) in each open Centrify zone.
Duplicate role assignment containers in computer	Check for computers that have more than one location to store role assignment information.
Duplicate service principal names in forest	Check for duplicate service principal names across the entire forest. Service principal names are required to be unique within an Active Directory forest.
Duplicate SFU zones	Check for duplicate SFU zones that are set to manage the same NIS domain.
Duplicate users in zones	Check for duplicate UNIX user names or user identifiers (UIDs) in each open Centrify zone.
Duplicate zone default container	Check for duplicate Zones parent container objects in the Active Directory forest.
Empty computer roles	Check for computer roles that contain no computers or role assignments.
Empty profiles in hierarchical zones	Check for hierarchical zones that contain users or groups that have no profile data defined.
Empty zones	Check for zones that have no computers, users, or groups.
Foreign Security Principal Clean Up	Check for foreign security principal objects whose corresponding security principal has been removed.
Incomplete user UNIX data	Check for users with missing UNIX profile attributes or who are missing a primary profile. This analysis option checks the entire zone hierarchy for profiles with missing attributes and for users who have multiple profiles defined but don not have a primary profile. Users with an incomplete profile or a missing primary profile will not be able to log on even if they are assigned a role with login rights. Note that a profile can be incomplete at any level of the zone hierarchy as long as it is complete at the level where a computer is joined.

Select this option	To do this
Inconsistency in granting NIS server permissions	Check that there is a `zone_nis_servers` group in each zone that supports agentless authentication and that the group contains all the NIS servers that have been defined for the zone. The `zone_nis_servers` group is required to assign permissions to managed computers that act as NIS servers, and should not be manually deleted or modified. This option checks that the group exists and includes all of the computers acting as NIS servers to ensure data integrity.
Inconsistent computer object names	Check for discrepancy between the DNS name for a computer in Active Directory and its Centrify computer profile name.
Insufficient permission for agent version update	Check whether the computer object in Active Directory has sufficient permission to update the version number property of the Centrify UNIX agent in the computer's `serviceConnectionPoint` object. If the computer object does not have permission to change this property, the version number cannot be displayed.
Insufficient permission for OS version update	Check whether the computer object in Active Directory has sufficient permission to update the version number property of the operating system in the computer's `serviceConnectionPoint` object. If the computer object does not have permission to change this property, the operating system version number cannot be displayed.
Invalid right assignments	Check whether an invalid right has been assigned to a role. This error occurs if a right has been added to a role and subsequently the right becomes invalid. Generally, a right becomes invalid if it is edited with a third-party tool, such as ADSI Edit, and an attribute is set to an invalid value. For example, Access Manager creates Active Directory objects of type `msDS-AZOperation` for command- and PAM-application- rights, and assigns a HEX value to the `msDS-AzOperationId` attribute of these objects. The range of reserved values for this attribute is as follows: ■ Command: (HEX) `0500,0000 – 05FF,FFFF` ■ PAM application: (HEX) `0200,0000 – 02FF,FFFF` If this attribute is set to a value that is out of the reserved range, the right will be invalid and will no longer appear in Access Manager. If the right has been assigned to a role, the Analyze Invalid right assignment check returns an error. You can select the error in the **Analysis Results** node and use the Action menu to delete it from the role if you wish.

Select this option	To do this
Invalid role assignments	Check whether invalid role assignments exist in the zone. In most cases, invalid role assignments occur when a role assignment is defined for a computer account and the computer leaves a zone without cleaning up role-assignment objects.

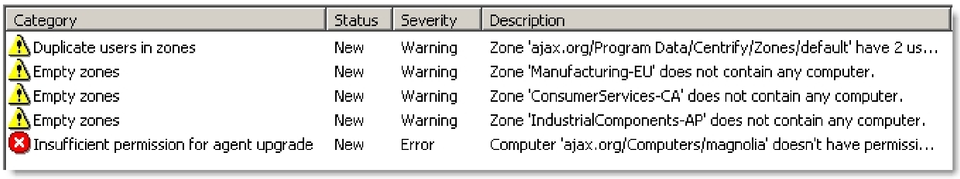
Invalid role assignments (DZ V2)	Check for role assignments that contain multiple roles or multiple users.
	In most cases, this error only occurs if you are using third-party tools to edit role assignments. Centrify tools prevent you from creating invalid role assignments.
	Note that a role assignment consists of a single user and a single role. To assign multiple roles to a user, you create multiple role assignments, which are stored in the form of user@domain role/sourceZone; for example:
	`qa1@acme.com login/engineering`
	`qa1@acme.com vi_power/engineering`
	`qa1@acme.com test/engineering`

Orphan child zones	Check for child zones that have an invalid parent zone. The information identifying the parent-child zone relationship is stored in the child zone in the form of a HEX string and the name of the domain to which the parent zone belongs. If this identifier is deleted, or changed to an invalid format, or if the parent zone is deleted but the child zone remains in the domain, Analyze (Orphan child zones) returns an error. Note that this error typically occurs only if you use third-party tools to edit zone objects in Active Directory. If you delete a parent zone using Centrify tools, child zones are deleted as well.
Orphan role assignments	Check for role assignments that consist of a non-existent role or user, or that do not contain a role or user. In most cases, this error only occurs if you are using third-party tools to edit Centrify objects in Active Directory. If you delete a role or user using Centrify tools or using Active Directory Users and Computers, the role assignment will be deleted as well (the change will be visible after you refresh the display) and Analyze will not return an error.
Orphan zone data objects and invalid data links	Check for zone data that have no corresponding Active Directory objects or have invalid links to Active Directory objects. For example, if you delete an Active Directory user but do not remove the profile for this user in a zone, the zone profile becomes an orphan and is flagged as such by this option.

Select this option	To do this
Restricted roles	Check for roles that have been assigned commands that cannot be executed. When rights are created, they can be defined to run in a restricted-shell role, in an enhanced role (with dzdo), or with both. If a command that has not been defined to run in a restricted-shell is added to a restricted-shell role, this check returns an error.
Zone created under another zone	Check for zone information created in another zone's parent container. Note that this check does not look at hierarchical zones because it is expected that child zones are physically contained in their parent zone.
Zone information in old format	Check for zone information stored in an obsolete Centrify zone format.
Zoneless computers	Check for computers that do not belong to any zone.

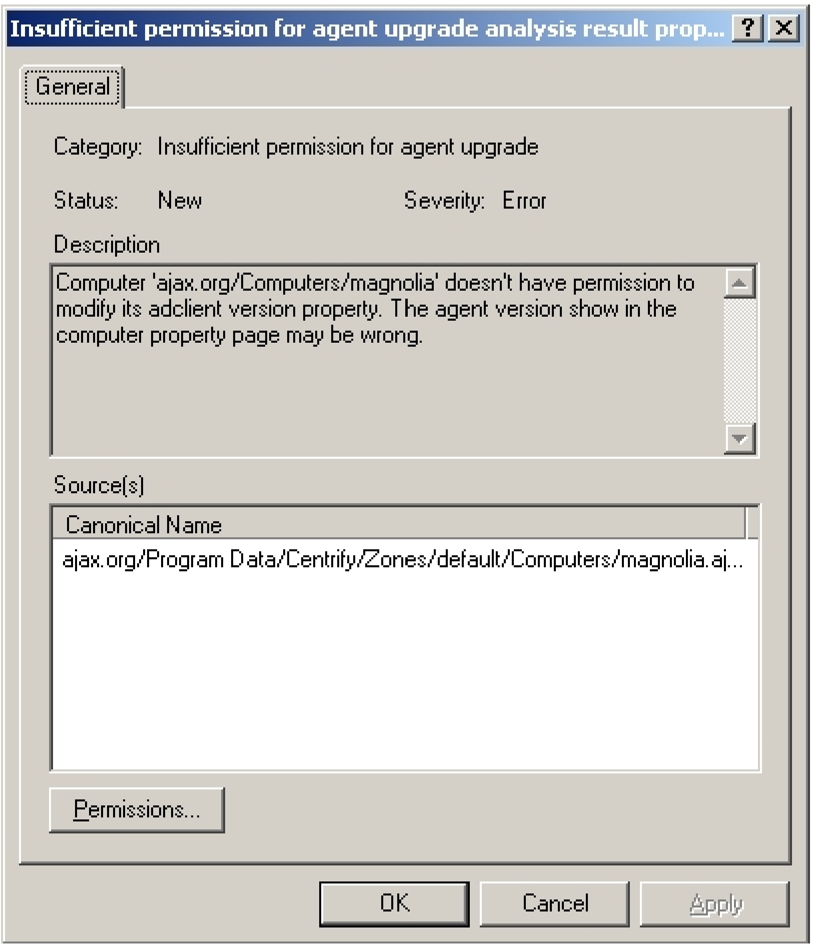
4. Review the result summary, then click **Finish**.

5. If the result summary indicates any issues, you can view the details by selecting **Analysis Results** in the console tree and viewing the information listed in the right pane. For example:



Category	Status	Severity	Description
⚠ Duplicate users in zones	New	Warning	Zone 'ajax.org/Program Data/Centrify/Zones/default' have 2 us...
⚠ Empty zones	New	Warning	Zone 'Manufacturing-EU' does not contain any computer.
⚠ Empty zones	New	Warning	Zone 'ConsumerServices-CA' does not contain any computer.
⚠ Empty zones	New	Warning	Zone 'IndustrialComponents-AP' does not contain any computer.
⊗ Insufficient permission for agent upgrade	New	Error	Computer 'ajax.org/Computers/magnolia' doesn't have permissi...

For additional information, select the warning or error, right-click, then select **Properties**. For example:



Insufficient permission for agent upgrade analysis result prop...
General

Category: Insufficient permission for agent upgrade

Status: New

Severity: Error

Description

Computer 'ajax.org/Computers/magnolia' doesn't have permission to modify its adclient version property. The agent version show in the computer property page may be wrong.

Source(s)

Canonical Name
ajax.org/Program Data/Centrify/Zones/default/Computers/magnolia.aj...

Permissions...
OK
Cancel
Apply

Common scenarios that generate analysis results

For most organizations, it is appropriate to check the data integrity of the Active Directory forest on a regular basis. Although running the Analyze command frequently may not be necessary for small networks with few domain controllers, there are several common scenarios that you should consider to determine how often you should check the forest for potential problems. The most likely reasons for data integrity issues stem from:

- Multiple administrators performing concurrent operations.
- Administrators using different domain controllers to perform a single operation.
- Replication delays that allow duplicate or conflicting information to be saved in Active Directory.
- Insufficient permissions that prevent an operation from being successfully completed.

- Network problems that prevent an operation from being successfully completed.
- Partial or incomplete upgrades that result in inconsistency of the information stored in Active Directory.
- Using ADEdit rather than the Console to create, modify, or delete zone objects, which may lead to problems, such as inadvertently creating a circular zone structure or an empty profile.
- Using third-party tools, such as ADSI Edit, to edit objects directly in Active Directory, which may lead to corrupted or invalid zone objects.

Running Analyze periodically helps to ensure the issues these scenarios can cause are reported in the Analysis Results, so you can take corrective action.

Responding to analysis results

Depending on the type of warning or error generated in the Analysis Results, you might be able to take corrective action or access additional information by right-clicking a result, then selecting an appropriate action. For example, if a computer account lacks the permission required to update Active Directory with the operating system version currently installed, you can right-click the warning in the Analysis Result then select **Grant computer the rights to modify operating system properties**.

If right-clicking a result does not provide a responsive action, you should use Access Manager or ADEdit to correct the issue.

The following table describes the warnings and errors you may see in the Analysis Results after running the Analyze wizard and how to resolve potential issues.

Result	Responsive action
If there are any computers joined to multiple zones, an error is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if an administrator runs `adleave` with the `--force` option then runs `adjoin` to join the computer to a different domain without removing the old computer profile from Active Directory. You should identify the appropriate zone for the computer, then use the Access Manager console to delete the computer profile from any additional zones.
If the parent-child relationship of any zones is circular, an error is displayed.	Break the circular relationship.
If there are any duplicate groups in a zone, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate group profile to be added to a zone. For example, if two administrators add the same group to a zone using different domain controllers, there will be duplicate group profiles after the domain controllers complete replication. You should use the Access Manager console or ADSI Editor to delete the duplicate group profiles from the zone.
If any duplicate service principal names (SPNs) are found for users or computers in the forest, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. Right-click the warning and click **Properties** to identify the duplicate SPN. Open the account properties for the user or computer and modify or remove the duplicate servicePrincipalName value. Alternatively, run the `adjoin` command with the `-d` or `--forceDeleteObjWithDupSpn` option. See the `adjoin` man page for additional information.

Result	Responsive action
If there are any duplicate users in a zone, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate user profile to be added to a zone. For example, if two administrators add the same user to a zone using different domain controllers, there will be duplicate user profiles after the domain controllers complete replication. You should use the Access Manager console or ADSI Editor to delete the duplicate user profiles from the zone.
If more than one Centrify SFU zone is found in the forest, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. Because an SFU zone is associated with an Active Directory SFU schema extension, there should be a maximum of one SFU zone in an Active Directory forest. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate. You should use the Access Manager console or ADSI Editor to delete any duplicate SFU zones.
If a duplicate default parent container for zones is found, a warning is displayed.	No responsive action can be taken directly within the Analysis Results for this issue. In general, this issue only occurs if multiple administrators perform concurrent operations or there are replication delays that allow a duplicate default container for new zones. Having more than one default parent container for zones can result in an unexpected default value in the Create New Zone wizard. You should use the ADSI Editor to delete any duplicate Zones parent containers from the forest.
If a computer role does not have any member computers or role assignments, a warning is displayed.	If the computer role has no member computers, right-click the warning in the Analysis Results, then select **Add computers** to add computers, or Delete Computer Role to remove the computer role. If a computer role has computer members but no role assignments, the only available response from the Analysis Results zone is to delete the computer role. You can, however, select the computer role in the Console, and add role assignments to its Role Assignments node.
If a user or group profile has been added to a zone but has no attributes defined, an error message is displayed.	Right-click the warning in the Analysis Results, then select **Delete empty profile** to delete the profile from the zone, or **Modify profile** to define one or more attributes for the user or group.

Result	Responsive action
If any zone does not contain users, groups, or computers, a warning is displayed for each type of object. For example, if a zone has computers and groups, but no users, only the user warning is displayed for that zone.	No responsive action can be taken directly within the Analysis Results for these issues. In general, this issue occurs early in a deployment before you have populated zones. You should use the Access Manager console to add missing objects to the zone. If the empty zone is not a valid zone, right-click the zone and select **Delete**.
If one or more secondary profiles are found for a user but no primary profile is found, a warning message is displayed.	Right-click the warning in the Analysis Results, then select **Promote secondary profile to primary** to select a secondary profile you want to make the primary profile for the user.
If a user's UNIX profile is incomplete in the entire zone hierarchy, a warning message is displayed.	Right-click the warning in the Analysis Results, then select **Modify zone profile** to define additional attributes to complete the user's profile.
If the Active Directory group `zone_nis_servers` is not found in a zone configured for agentless authentication, an error is displayed.	Right-click the error in the Analysis Results, then select **Create NIS servers group** to create the `zone_nis_servers` group for agentless authentication. Note that your account must have permission to create this object for the operation to be successful.
If the membership of the `zone_nis_servers` group is not consistent with the computers authorized as NIS servers, a "Membership inconsistent" error is displayed.	Right-click the error in the Analysis Results, then select **Fix group membership** to modify the membership list for the `zone_nis_servers` group.
If a zone is configured to support agentless authentication and the `zone_nis_servers` group exists but does not contain all computers in the zone, an informational alert is displayed.	No responsive action can be taken directly within the Analysis Results for these issues. You should verify that all of the computers you want to use as NIS servers in the zone are configured to allow agentless authentication.
If there is a discrepancy between the DNS name in AD and the Centrify computer profile name, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Fix group membership** to

Result	Responsive action
If a computer account does not have permission to write to the `keywords` attribute, an error is displayed.	Right-click the error in the Analysis Results, then select **Grant permission to computer account** to update the permissions on the computer account object.
If a computer account does not have permission to modify operating system properties, a warning is displayed.	Right-click the error in the Analysis Results, then select **Grant computer permission to modify operating system properties** to update the permissions on the computer account object.
If a right for a role is invalid, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Delete Right** to delete the right from the role.
If a role assignment is invalid, a warning message is displayed.	
If multiple roles are assigned to a user, a warning message is displayed.	
If a child zone has an invalid parent zone, an error message is displayed.	
If an object has no parent object, a warning message is displayed.	
If a restricted-shell role is assigned a right that cannot be run in a restricted shell, a warning message is displayed.	Right-click the error in the Analysis Results, then select **Delete Commands** to remove the commands from the role, or select **Allow running in restricted role** to allow running the command in the restricted role.
If a zone was created using the version 2.x console and includes a Private Groups container, a warning is displayed.	If any computers in the zone are running version 2.x or 3.x agents, you should ignore this warning to ensure compatibility for those agents. If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove privateGroupCreation attribute** to update the zone format.

Result	Responsive action
If a computer profile was created using the version 2.x console, the warning “Unix computer is in old format” is displayed.	If any computers in the zone are running version 2.x or 3.x agents, you should ignore this warning to ensure compatibility for those agents. If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy and unix_enabled attribute** to update the computer profile in the zone.
If a group profile was created using the version 2.x console, the warning “Unix group is in old format” is displayed.	If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy attribute** to update the group profile in the zone.
If a user profile was created using the version 2.x console, the warning “Unix user is in old format” is displayed.	If all of the agents in the zone have been upgraded, you can right-click the warning in the Analysis Results, then select **Remove managedBy and app_enabled attribute** to update the user profile in the zone.
If a computer, group, or user profile exists, but no corresponding Active Directory computer, group, or user object is found, the warning “Orphan UNIX data object” is displayed.	In general, this issue occurs if an administrator removes an Active Directory computer, group, or user object manually using ADSI Editor or Active Directory Users and Computers but the corresponding data is not removed for the UNIX profile. Right-click the warning in the Analysis Results, then select **Remove orphan profile** to remove all of the UNIX properties associated with the orphan profile.
If a computer, group, or user profile has inconsistent links, an informational “Inconsistent links” alert is displayed.	Computer, group, and user profiles are associated with Active Directory computer, group, and user objects through either the `managedBy` attribute (agent version 2.x) or a `parentLink` value in the `keywords` attribute (agent version 3.x and later). If the links refer to different Active Directory objects, you will see this alert. Right-click the alert in the Analysis Results, then select **Overwrite with the active link** to remove outdated links.
If a computer, group, or user profile does not have a `parentLink` value defined, a “Missing parentLink” warning is displayed.	Right-click the warning in the Analysis Results, then select **Missing parentLink** to add the parentLink value to the keywords attribute.

Result	Responsive action
If the parent container for a zone is another zone object, an error is displayed.	No responsive action can be taken directly within the Analysis Results for these issues. You should move the zone to another parent container or delete and recreate the zone in a different location.
The computer `ObjectName` contains Centrify information but it is not in a zone.	Right-click the warning in the Analysis Results, then select **Move to Zone** to search for and select the zone you want to place the computer in.

Configuring logging for the agent

By default, the Centrify UNIX agent logs errors, warnings and informational messages in the UNIX `syslog` and `/var/log/messages` files along with other kernel and program messages. Although these files contain valuable information for tracking system operations and troubleshooting issues, occasionally you may find it useful to activate agent-specific logging and record that information in a log file.

To enable logging on the Centrify UNIX agent

1. Log in as or switch to the `root` user.
2. Run the `addebug` command:

   ```
   /usr/share/centrifydc/bin/addebug on
   ```

   Note: You must type the full path to the command because `addebug` is not included in the path by default.

   Once you run this command, all of the Centrify agent activity is written to the `/var/log/centrifydc.log`file. If the `adclient` process stops running while you have logging enabled, the `addebug` program records messages from PAM and NSS requests in the `/var/centrifydc/centrify_client.log` file. Therefore, you should also check that file location if you enable logging.

For performance and security reasons, you should only enable logging when necessary, for example, when requested to do so by Centrify Support, and for short periods of time to diagnose a problem. Keep in mind that sensitive information may be written to this file and you should evaluate the contents of the file before giving others access to it.

When you are ready to stop logging activity, run the `addebug off` command.

Setting the logging level

You can define the level of detail written to the log by setting the `log` configuration parameter in the Centrify configuration file:

```
log: level
```

With this parameter, the log level works as a filter to define the type of information you are interested in and ensure that only the messages that meet the criteria are written to the log. For example, if you want to see warning and error messages but not informational messages, you can change the log level from `INFO` to `WARN`. By changing the log level, you can reduce the number of messages included in the log and record only messages that indicate a problem. Conversely, if you want to see more detail about system activity, you can change the log level to `INFO` or `DEBUG` to log information about operations that do not generate any warnings or errors.

You can use the following keywords to specify the type of information you want to record in the log file:

Specify this level	To log this type of information
`FATAL`	Fatal error messages that indicate a system failure or other severe, critical event. In addition to being recorded in the system log, this type of message is typically written to the user's console. With this setting, only the most severe problems generate log file messages.
`ERROR`	System error messages for problems that may require operator intervention or from which system recovery is not likely. With this setting, both fatal and less-severe error events generate log file messages.
`WARN`	Warning messages that indicate an undesirable condition or describe a problem from which system recovery is likely. With this setting, warnings, errors, and fatal events generate log file messages.
`INFO`	Informational messages that describe operational status or provide event notification.

Logging for Access Manager

Although most logging activity focuses on the actions of the Centrify agent, you can also enable or disable logging for the Access Manager console and configure the types of messages to record in the log file by selecting options in Access Manager.

To configure logging for operations handled through the Access Manager console:

1. Open Centrify Access Manager.
2. In the console tree, select **Centrify Access Manager**, right-click, then click **Options**.
3. Click the **Log Settings** tab, select the type of messages to log, then click **OK**.

If you enable logging, the log file is located by default in the `C:\Users\user\AppData\Roaming\Centrify\DirectControl` folder and is updated as you perform different operations in the Access Manager console.

Logging to the circular in-memory buffer

If the Centrify UNIX agent's `adclient` process is interrupted or stops unexpectedly, a separate watchdog process (`cdcwatch`) automatically enables an in-memory circular buffer that writes log messages passed to the logging subsystem to help identify what operation the `adclient` process was performing when the problem occurred. The in-memory buffer is also mapped to an actual file, so that if there's a system crash or a core dump, the last messages leading up to the event are saved. Messages from the in-memory circular buffer have the prefix `_cbuf`, so they can be extracted from a core file using the `strings` command.

The in-memory circular buffer allows debug-level information to be automatically written to a log file even if debugging is turned off. It can be manually enabled by restarting the `adclient` process with the `-M` command line option. The default size of the buffer is 128K, which should be sufficient to log approximately 500 messages. Because enabling the buffer can impact performance, you should not manually enable the circular buffer or modify its size or logging level unless you are instructed to make the changes by Centrify Support.

Collecting diagnostic information

You can use the `adinfo` command to display or collect detailed diagnostic and configuration information for a local UNIX computer. Options control the type of information and level of detail displayed or collected. The options you are most likely to use to collect diagnostic information are the `--config`, `--diag`, or `--support` options, which require you to be logged in as `root`. You can redirect the output from any `adinfo` command to a file for further analysis or to forward information to Centrify Support.

For more information about the options available and the information returned with each option, see the `man` page for `adinfo`.

To display the basic configuration information for the local UNIX computer, you can type:

```
adinfo
```

If the computer has joined a domain, this command displays information similar to the following:

```
Local host name:   magnolia
Joined to domain:  ajax.org
Joined as:         magnolia.ajax.org
Current DC:        ginger.ajax.org
Preferred site:    Default-First-Site-Name
Zone:              ajax.org/Centrify/Zones/corporate
Last password set: 2006-12-28 14:47:57 PST
CentrifyDC mode:   connected
Licensed Features  Enabled
```

Working with domain controllers and DNS servers

Centrify agents are designed to perform the same set of DNS lookup requests that a typical Windows workstation performs to find the nearest domain controller for the local site. The DNS lookup request enables the Centrify UNIX agent to find domain controllers as they become available on the network or as the computer is relocated to another network location where different domain controllers are present. Centrify agents also use DNS to find the Kerberos service providers and the global catalog service providers for the Active Directory forest.

In a typical Windows environment, the DNS server role is updated dynamically to contain the service locater (SRV) DNS entries for Active Directory’s LDAP, Kerberos, and global catalog services, so this information is available for

Centrify agents to use. However, there are some configurations of DNS that might not provide all of the SRV records for the set of domain controllers that provide Active Directory service to the enterprise. You may also run into problems if DNS for the enterprise runs on UNIX servers that cannot locate your Active Directory domain controllers. The next sections describe how you can adjust DNS or Centrify agent to ensure they work together properly in your environment.

Configuring the DNS server role on Windows

One of the most common scenarios for running DNS in an environment with Active Directory is to add the DNS server role to a Windows domain controller or another Windows server.

If you are already using DNS in Active Directory and dynamically publishing DNS service records, no additional configuration should be necessary. If you are using DNS in Active Directory but have disabled dynamic updates, you should change the configuration for the DNS server role to allow dynamic updates. Making this change will allow Centrify agents to properly locate domain controllers in the site and select an appropriate new domain controller if a connection to its primary domain controller is lost or the managed computer is moved to a new location on the network.

Configuring DNS running on UNIX servers

If your environment is configured to use UNIX-based DNS servers instead of Active Directory-based DNS servers and the UNIX system is configured to use DHCP, the `nameserver` entry in `/etc/resolv.conf` file is set automatically to point to a DNS server.

If this DNS server is aware of the Active Directory domain you want to join, no further changes are needed. If the DNS server identified as a `nameserver` in the `/etc/resolv.conf` file is not aware of the domain you are trying to join, for example, because you are using a test domain or a separate evaluation environment, you need to either disable DHCP or manually set the location of the Active Directory domain controller in the Centrify configuration file.

• • • • • •

Checking whether DNS can resolve the domain controller

In most cases, you can verify whether a UNIX computer can locate the domain controller and related services by running the `ping` command and verifying connectivity to the correct Active Directory domain controller or by checking the `nameserver` entry in the `/etc/resolv.conf` file. This `nameserver` entry should be the IP address of one of the domain controllers in the domain you want to join.

If the `ping` command is successful, it indicates the DNS server is aware of the Active Directory domain you want to join and no further changes are needed. If the `ping` command is not successful, you will need to take further action to resolve the issue.

Resolving issues in locating Active Directory domain controllers

If the UNIX computer cannot find the Active Directory domain controller, there are several ways you can resolve the issue. Depending on your environment and specific situation, you should consider doing one of the following:

- Set up DNS on the target Active Directory domain controller and the manually configure the `nameserver` entry in the `/etc/resolv.conf` file to use that domain controller as described in Setting up DNS service on a target domain controller.
- Set the Centrify configuration file to manually identify the domain controllers you want to use as described in Setting the domain controller in the configuration file.

Setting up DNS service on a target domain controller

One of the simplest ways to ensure that the UNIX computers can locate the Active Directory domain controller and related services is to use the DNS service on the Active Directory domain controller as a DNS slave to the enterprise DNS servers. You can do this is by configuring the DNS server role on the Active Directory domain controller, then specifying that domain controller in the UNIX computer's `/etc/resolv.conf` file. You can then add a forwarder to the local DNS on the domain controller that will pass on all lookups that it cannot satisfy to an enterprise DNS server.

This configuration does not require any changes to the enterprise DNS servers. Any look up request from the domain controller is simply a query from another computer in the enterprise. However, the UNIX computers configured to use this slave DNS service will receive the appropriate Service Location (SRV) records and global catalog updates for the Active Directory domain controller. In addition, the DNS service on the domain controller can be configured to forward requests to the enterprise DNS servers so those requests can be answered when the local DNS service cannot respond.

Adding a DNS server role to an Active Directory domain controller

The specific steps for adding the DNS server role to a domain controller depend on the version of Windows Server you use. In most cases, you can use an administrative tool, such as Server Manager, to add roles. Follow the instructions displayed in the wizard to add the **DNS Server** server roles, configure the DNS server lookup zones, select the **Allow both nonsecure and secure dynamic updates** option.

After you have configured the DNS server role on the domain controller, the computer uses the local DNS server as its primary DNS server.

Configuring UNIX to use DNS service on the target domain controller

Once you have configured the DNS service to contain the required Active Directory entries, you simply need to modify the UNIX computer to send all DNS lookup requests to the newly configured DNS server.

To configure the UNIX computer to use the new DNS server:

1. Open the `/etc/resolv.conf` file.
2. Set the IP address of the `nameserver` entry to the IP address of the DNS server on the Active Directory domain controller you just configured.

Setting the domain controller in the configuration file

If you are not able to use DNS to locate the Active Directory domain controllers on your network, you can manually specify one or more domain controllers in the Centrify configuration file.

To manually specify a domain controller, add the following entry to the Centrify configuration file, `/etc/centrifydc/centrifydc.conf`:

```
dns.dc.domain_name: server_name [server_name ...]
```

For example, if you want to ensure the Centrify agent uses the domain mylab.test and the domain controller named dc1.mylab.test, you could add the following line to the /etc/centrifydc/centrifydc.conf file:

```
dns.dc.mylab.test: dc1.mylab.test
```

> **Note:** You must specify the name of the domain controller, not its IP address. In addition, the domain controller name must be resolvable using either DNS or in the local /etc/hosts file. Therefore, you must add entries to the local /etc/hosts for each domain controller you want to use if you are not using DNS or if the DNS server cannot locate your domain controllers.

To specify multiple servers for a domain, use a space to separate the domain controller server names. For example:

```
dns.dc.mylab.test: dc1.mylab.test dc2.mylab.test
```

The Centrify agent will attempt to connect to the domain controllers in the order specified. For example, if the domain controller dc1.mylab.test cannot be reached, the agent will then attempt to connect to dc2.mylab.test.

If the global catalog for a given domain is on a different domain controller, you can add a separate dns.gc.domain_name entry to the configuration file to specify the location of the global catalog. For example:

```
dns.gc.mylab.test: dc3.mylab.test
```

You can add as many domain and domain controller entries to the Centrify configuration file as you need. Because the entries manually specified in the configuration file override any site settings for your domain, you can completely control the Centrify UNIX agent's binding to the domains in your forest through this mechanism.

> **Note:** In most cases, you should use DNS whenever possible to locate your domain controllers. Using DNS ensures that any changes to the domain topology are handled automatically through the DNS lookups. The settings in the configuration file provide a manual alternative to looking up information through DNS for those cases when using DNS is not possible. If you use the manually-defined entries in the configuration file and the domain topology is changed by an Active Directory administrator, you must manually update the location of the domains in each configuration file.

Using the fixdns script

The Centrify agent includes a `fixdns` script that you can use to inspect your environment and make the necessary configuration file changes for you.

To run this script, you need to specify the domain controller name and IP address:

```
fixdns domain_controller_name IP_address
```

For example if you intend to join the domain `mytest.lab` and the domain controller for that domain is `dc1.mytest.lab` and its address is `172.27.20.1`, you would run the following command:

```
fixdns dc1.mytest.lab 127.27.20.1
```

The `fixdns` script will then make the necessary changes to the `/etc/hosts` and the Centrify configuration file.

> **Note:** This script does not update the `/etc/resolv.conf` file. If the script cannot locate the domain controller using the existing `/etc/resolv.conf` settings, it will assume that you want to use settings from the configuration file.

What the Centrify DNS subsystem provides

Centrify provides a DNS subsystem that bypasses the local DNS resolver to address common issues that occur with many local DNS resolvers. These common issues for local DNS resolvers include:

- Degraded performance when connecting to a slow DNS server or when attempting to use dead DNS servers.
- Degraded performance when reacquiring a DNS server that went offline and has come back online.
- Degraded performance related to DNS timeouts.
- Platform-related DNS idiosyncrasies, such as MDNS, appending.LOCAL suffixes, and so on.

The Centrify DNS subsystem performs the following functions:

- Looks up hosts by name.
- Looks up hosts by IP address.

- Queries DNS service location records (SRV) to discover the domain controllers that support Active Directory services including KDC, KPASSWD, LDAP and the global catalog.

Resolving a host name or IP address

When the DNS client subsystem receives a DNS requests, it attempts to resolve the host name or IP address by first checking the `/etc/hosts` file. If the file contains a valid entry to resolve the specified host name or IP address, the DNS client subsystem processes the DNS request.

Entries in `/etc/hosts` must be in the following format:

```
IPv4_address hostname alias alias ...
```

where:

- IPv4_address must be in the first position
- hostname is a fully-qualified domain name and must be in the second position.
- aliases are optional and follow the address and hostname entries.

For example:

```
192.169.147.135 ginger.acme.com ginger
```

Note: Service (SRV) record queries cannot be satisfied from the `/etc/hosts` file.

If resolution by `/etc/hosts` is unsuccessful, the DNS subsystem attempts to select a DNS server that can be used to resolve the host name or IP address (as described in the next section, Selecting a DNS server).

Selecting a DNS server

If unable to resolve a host name or IP address by finding an entry in the `/etc/hosts` name (as described in the previous section, Resolving a host name or IP address), the Centrify DNS subsystem attempts to find a DNS server to resolve the host name or IP address, as follows:

- It checks for a working DNS server that has already been selected (cached in memory and stored in `/var/centrify/kset.dns.server`), and if available, uses it.

- If a working DNS server is not already selected, it checks `/etc/resolv.conf` for configured DNS servers, and if populated, selects the fastest one from the list.

If no working DNS servers are found, the request fails.

At this point, DNS is considered down, and the Centrify DNS subsystem waits for the interval specified by the `dns.dead.resweep.interval` (default is 60 seconds), before attempting again to find a DNS server.

Specifying DNS-related parameters

Parameters in the Centrify configuration file control many aspects of Centrify DNS subsystem operation. Although you can set any of these parameters, the default settings should provide you with optimal DNS operation. See the Configuration and Tuning Reference Guide for details about any of these parameters.

The DNS subsystem periodically checks in the background to see if a DNS server that is faster than the currently selected one is available. The `dns.alive.resweep.interval` parameter determines how often this background check occurs; the default value is one hour (3600 seconds).

When a DNS server is selected, its address is stored in the `kset.dns.server` file, and it is used for all DNS requests until one of the following occurs:

- The selected server stops responding.
- A new server sweep discovers a faster DNS server and replaces it.
- The `adclient` process is stopped and restarted, which triggers a sweep for a new DNS server.
- The specified server is no longer in the list of servers in `/etc/resolv.conf`.

For the sweep, the `dns.sweep.pattern` parameter determines the probe pattern that is used to find a live DNS server; that is, it sets the protocol to use (TCP or UDP) and the amount of time to wait for a response. By default, this parameter specifies both a TCP and UDP probe.

The `dns.timeout` and `dns.udp.retries` parameters determine the amount of time to wait, and how often to re-send a request when the current server does not respond to a request. If the current server does not respond to a request within the specified time out period, it is considered down and Centrify looks for a different server. If it cannot find a live server, DNS is considered down, and the Centrify UNIX agent waits for the period of the `dns.dead.resweep.interval` parameter, 60 seconds by default, before performing a sweep to find a new server.

Filtering the objects displayed

For performance or security reasons, you might want to filter or limit the objects displayed in the Access Manager console. Depending on your environment, you might want to display more or less information by setting filter options. These filter settings enable you to control both the number and type of objects displayed. You should note, however, that these settings can affect the performance of the console.

To filter the objects listed in Access Manager:

1. Open Access Manager.
2. In the console tree, select **Access Manager**, right-click, then click **Options**.
3. Click the **Filter Settings** tab.
4. Select **Load all zones** to automatically open either all zones in the connected forest or all zones in a specific parent container.
    - If you select this option and **connected forest** all zones in the forest are opened automatically each time you start Access Manager. Selecting this option prevents you from opening or closing any zones manually. Depending on the number of zones you have, you might experience slower performance in the console if you select this option.
    - If you select this option and **container**, you can then click Browse to search for a container from which to automatically load zones. Selecting this option prevents you from opening or closing any zones manually. Depending on the number of zones you have in the selected container, you might experience slower performance in the console if you select this option.
5. Select **Show disabled Active Directory accounts** to display disabled computer and user accounts or uncheck this option to hide disabled objects.

6. Select **Show orphans** to display all users, groups, and computers that have a UNIX profile or uncheck this option to hide all orphan profiles.

   Orphan profiles are the service connection points that no longer have a corresponding Active Directory object. Hiding or removing orphan profiles can improve console performance. For information about locating orphan profiles by running an analysis on the Active Directory forest, see Analyzing information in Active Directory.

7. Select **Show Auto Zone** to display the users, groups, and computers that have joined the Auto Zone or uncheck this option to hide Auto Zone information.

8. Set the **Maximum number of items to be displayed in the list** option to limit the total number of objects displayed in the console, up to total maximum allowed (65535).

   This setting applies to all of the objects displayed in Access Manager, including zones, computers, users, groups, pending users, pending groups, NIS maps, and all defined rights, roles, and role assignments. Lowering the maximum number of items displayed improves performance when browsing the listed items. Note that this setting does not affect the number of items you can define, only the number displayed.

9. Click **OK**.

Centrify Authentication Service issues on *NIX systems

Be aware of the following issue when working with Centrify Authentication Service on UNIX or Linux systems:

- The directory /var should not be NFS mounted or else DirectControlmay not work properly. (Ref: IN-90009)
- Please see KB-9092 for further details about using common UNIX commands with Centrify Privilege Elevation Service restricted shells.

Using Centrify commands for administrative tasks

This chapter provides an overview of the Centrify command-line interface and a list of the command-line programs you can execute locally on Centrify-managed computers.

How and when to use command-line programs

UNIX command-line programs are installed by default when you install the Centrify UNIX agent. The commands are typically installed in one of the following directories:

```
/usr/sbin, /usr/bin
/usr/share/centrifydc/bin
```

The Centrify agent includes a large number of command-line programs that enable you to perform a variety of administrative tasks directly from a UNIX shell or using a shell script. These command-line programs use the underlying `adclient` service library to perform important tasks on the computers you add to Active Directory domains. For example, there are commands that allow you to remove a computer from an Active Directory domain, change an Active Directory user's password, and return detailed diagnostic information about the operations of a host computer.

You can use command-line programs interactively or in shell scripts when you must take action directly on a Centrify-managed computer, or when taking action from a managed computer is most convenient. For example, individual users can use a command-line program to change their Active Directory password from a login shell without logging on to a Windows computer.

Some command-line programs perform specific tasks that you will only use infrequently or under specific conditions. Other programs perform common administrative tasks that you are likely to use repeatedly.

The most commonly used programs include the following:

- The `adjoin` command is the first command you use to add a local computer to an Active Directory domain.
- The `adinfo` command display summary or detailed diagnostic and configuration information for a computer and its Active Directory domain.
- The `adpasswd` command allows you to change an Active Directory account password from a Centrify-managed computer.
- The `adgpupdate` command allows you to force group policies to be refreshed immediately.
- The `adleave` command allows you to remove a managed computer from its current Active Directory domain or from the Active Directory forest entirely.

Displaying usage information and man pages

To display a summary of usage information for any command-line program, type the command and the `--help` or `-h` option. For example, to see usage information for the `adleave` command, type:

```
adleave --help
```

The usage information includes a list of options and arguments, and a brief description of each option. For example, if you specify `adleave -h` on the command line, the command displays the command-line syntax and a list of the valid options you can use when you execute adleave commands, similar to the following:

```
usage: adleave [options]
options:
  -u, --user user[@domain] user name, default is administrator
  -p, --password pw        user password, prompts if absent
  -s, --server ds          domain server for leave operations
  -Z, --zoneserver ds      domain server for zone operations
                           useful if zone is in another domain
  -C, --noconf             do not restore PAM or NSS config
  -G, --nogp               do not restore Group Policy
  -f, --force              force local leave, no network activity
  -v, --version            print version information
  -h, --help               print this help information and exit
```

For more complete information about any command, you can review the information in the command's manual (`man`) page. For example, to see the manual page for the `adleave` command, type:

```
man adleave
```

Result codes used by multiple programs

Many Centrify command-line programs share a common set of result codes returned when an operation is successful or an error occurs. The following table lists the result codes that are reserved for use by Centrify command-line programs.

Result	Error name	Indicates
0	ERR_SUCCESS	Successful completion of the operation.
6	ERR_OTHERS	Miscellaneous errors occurred during the operation.
7	ERR_USAGES	Usage error occurred during the operation.
8	ERR_OP_ABORTED	Operation aborted by user.
9	ERR_ROOT_PRIV	Root privilege is required for the operation.
10	ERR_NOT_JOINED	Computer is not currently joined to any Active Directory domain.
11	ERR_ALREADY_JOINED	Computer is already joined to the current Active Directory domain.
12	ERR_JOINED_ANOTHER_DOMAIN	Computer is currently joined to another Active Directory domain.
13	ERR_ADCLIENT_DOWN	The `adclient` process is not running or not available.
14	ERR_ADCLIENT_DISCONNECTED	The `adclient` process is running in disconnected mode.
15	ERR_ADLCIENT_STARTUP	The `adclient` process failed to start.
16	ERR_DNS_TIMEOUT	The DNS server is not responding and may be down.
17	ERR_DNS_GENERIC	A generic DNS problem occurred during the operation.
18	ERR_INVALID_DOMAIN_NAME	The Active Directory domain name is incorrect or not found in DNS.
19	ERR_INVALID_LOGON	User name or password provided is not correct.
20	ERR_ACCOUNT_DISABLED	The account specified has been disabled.
21	ERR_ACCOUNT_EXPIRED	The account specified has expired.
22	ERR_ACCOUNT_EXISTS	The account specified already exists,
23	ERR_ACCOUNT_NOTFOUND	The account specified was not found in Active Directory.
24	ERR_PASSWORD_EXPIRED	The account password has expired.
25	ERR_ZONE_NOTFOUND	The zone cannot be found.

Result	Error name	Indicates
26	ERR_CONTAINER_ NOTFOUND	Invalid Active Directory container object.
27	ERR_INSUFFICIENT_ PERM	The account specified does not have permission to perform the operation.
28	ERR_CLOCK_SKEW	The time difference between system clocks is beyond the acceptable range.
29	ERR_COMPUTER_NAME	Invalid computer account.
30	ERR_CRED_INVALID	Invalid credentials.
31	ERR_SERVICE_TKT_ INVALID	Invalid service ticket.
32	ERR_POLICY_NOT_ MATCH	Policy not matched.
33	ERR_REJECT_CHG_ PASSWD	Password change rejected.
34	ERR_WORKSTATION_ DENY	Workstation denied.
35	ERR_NOT_FIND_USER	No matching user found.
36	ERR_NOT_FIND_GROUP	No matching group found.
37	ERR_NOT_CONNECT_ ADCLIENT	An attempt to open a connection to the `adclient` process failed.
38	ERR_ADLCIENT_STOP	Unable to stop the `adclient` process.
39	ERR_QUOTA_EXCEEDED	The user has exceeded the number of join operations allowed.
40	ERR_OPEN_FILE	The attempt to open a file failed.
41	ERR_READ_FILE	The attempt to read a file failed.
42	ERR_COPY_FILE	The attempt to copy a file failed.

For information about command-specific result codes, see the manual page for individual command-line programs.

Perform administrative tasks using commands

Most administrative tasks can be performed using Access Manager on a Windows computer or by using ADEdit commands or scripts from a Centrify-managed computer that has access to the Active Directory domain controller. In some cases, however, there are operations that you must or prefer to perform locally on a managed computer by executing command-line programs.

The command line programs allow you to perform administrative tasks—such as join or leave a domain or generate diagnostic information—directly in a UNIX shell. Many of the command-line programs require administrative privileges or must run using `root` to perform privileged operations. You can define command rights for these programs to grant permission to run them to other users.

The following table provides a summary of the command-line programs for access control and privilege management that are installed with the Centrify UNIX agent. For complete information about the options you can specify for any command, see the `man` page for that command.

This command	Enables you to do this
adcache	Clear the local cache on a computer. You can use this command to clear all cached information or a specific cache file. You can also use the command to check a cache file for a specific key value and to reclaim disk space.
adcheck	Check the operating system, network, and Active Directory connections to verify that a computer is ready to join an Active Directory domain.
adchzone	Move a joined computer from a classic zone to a hierarchical zone. Before moving a computer with this command, you must use `admigrate` to migrate the classic zone to a hierarchical zone.
adclient	Start, stop, or manage operations for the Centrify agent process on a local computer. In most cases, you should start and stop `adclient` using a startup script.
addebug	Start or stop detailed logging activity for the Centrify agent (`adclient`) process on a local computer. If you do not specify an option, the addebug command displays its current status, indicating whether logging is active or disabled. You must be logged in as `root` to run this command.
addbloader	Create a database file with zone information. You can then use the `adreport` command to generate reports from this file, or read it with standard tools.
addns	Update DNS records on an Active Directory-based DNS server in environments where the DHCP server cannot update DNS records automatically.
adfinddomain	Display the domain controller associated with the Active Directory domain you specify.
adfips	Enable or disable FIPS-compliant encryption. You must be logged in as `root` to run this command.
adfixid	Resolve UID and GID conflicts and change the ownership of a local user's files to match the user and group IDs defined for the user in Active Directory.

This command	Enables you to do this
adflush	Clear the cache on a local computer. Executing `adflush` with no options expires the domain controller and global catalog caches.
adgpupdate	Retrieve group policies from the Active Directory domain controller and apply the policy settings to the local computer and current user immediately.
adid	Display the real and effective UIDs and GIDs for the current user or a specified user.
adinfo	Display detailed Active Directory, network, and diagnostic information for a local computer. Options control the type of information and level of detail displayed.
adjoin	Add the local host computer to the specified Active Directory domain. You must log in as `root` to run the `adjoin` command.
adkeytab	Create and manage Kerberos key tables (`*.keytab` files) and coordinate changes with the Kerberos key distribution center (KDC) provided by Active Directory. The arguments required and options available depend on the operation you want to perform.
adleave	Remove the local host computer from its current Active Directory domain. You must log in as `root` to run the `adleave` command.
adlicense	Enable or disable licensed features on a local computer. You must log in as `root` to run the `adlicense` command.
admanagelocal	Display currently managed local accounts, status of local account management, and force a foreground sync of local accounts.
admigrate	Migrate information from a classic zone to a hierarchical zone. You can migrate a classic zone to a new peer hierarchical zone, or you can specify a parent zone for the migration.
adobfuscate	Obscure sensitive information, such as email addresses, host names, and user names, that might be recorded in a log file before sending the file to Centrify for analysis. You must create a pattern file to use with this command. The command reads the pattern file and replaces items matching the patterns specified with generic values.
adpasswd	Change the password of the user executing the command or change the password of another Active Directory user.
adquery	Query Active Directory for information about users and groups from the command line on a Centrify-managed computer. This command is provided for backward compatibility. In most cases, you should use `adedit` commands or scripts to perform administrative tasks in Active Directory from Linux or UNIX computers.
adreload	Force the Centrify agent process (`adclient`) to reload the configuration properties in the `/etc/centrifydc.conf` file and in other files in the `/etc/centrifydc` directory.

This command	Enables you to do this
adreport	Generate user, computer, command, and role assignment reports for a zone. You must run the `addbloader` command to create a database containing information about a zone before you can run this command to generate a report.
adrmlocal	Report and remove local user names that duplicate Active Directory user names.
adsendaudittrailevent	Specify where to send audit trail events. You can choose to send audit trail events to the `syslog` facility, the Centrify auditing service, or both.
adsetgroups	View or change the list of groups available for the current user.
adsmb	Perform file operations, such as get a file, write a file, or display the contents of a directory using the Centrify `smb` stack.
adupdate	Update user and group account information from the command line on Centrify-managed computer. This command is provided for backward compatibility. In most cases, you should use `adedit` commands or scripts to perform administrative tasks in Active Directory from Linux or UJNIX computers.
dzdo	Execute a privileged command as `root` or another specified user. You must be assigned a role that grants privileged command rights to use this command.
dzedit	Edit a file as `root` or another user.
dzinfo	Display detailed information about the configuration of rights and roles for one or more specified users on the local computer. If you do not specify a user, the command returns information for the currently logged on user.
dzsh	Run commands in a restricted environment shell. This shell is a customized Bourne shell that provides environment variables, job control, command history, and access to specific commands defined by roles.
ldapadd	Open a connection to the Active Directory domain controller or another LDAP server to add new entries.
ldapcompare	Open a connection to the specified Active Directory domain controller or another LDAP server to compare LDAP entries. You can use this command to determine whether a specified entry has a particular attribute-value combination. The only information returned is whether the comparison evaluated to true or false. No other information about the entry is provided.
ldapdelete	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to delete the specified entry or entries.

This command	Enables you to do this
ldapmodify	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to modify the specified entry or entries.
ldapmodrdn	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to move or rename the specified entry or entries.
ldapsearch	Open a connection to the specified Active Directory domain controller or another LDAP server using the provided distinguished name and password to locate and retrieve the specified entry or entries.
nisflush	Clear the Centrify Network Information Service cache on a local computer, or restart the service without flushing the cache. You must be logged in as the `root` user to run this command.