Centrify Zero Trust Privilege Services: Authentication Service, Privilege Elevation Service, and Audit and Monitoring Service

Access Control and Privilege Management Scripting Guide

December 2020 (release 2020.1)

Centrify Corporation

Legal Notice

This document and the software described in this document are furnished under and are subject to the terms of a license agreement or a non-disclosure agreement. Except as expressly set forth in such license agreement or non-disclosure agreement, Centrify Corporation provides this document and the software described in this document "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Some states do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of Centrify Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of Centrify Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Centrify Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

Centrify, DirectControl, DirectAuthorize, DirectAudit, DirectSecure, DirectControl Express, Centrify for Mobile, Centrify for SaaS, DirectManage, Centrify Express, DirectManage Express, Centrify Suite, Centrify User Suite, Centrify Identity Service, Centrify Privilege Service and Centrify Server Suite are registered trademarks of Centrify Corporation in the United States and other countries. Microsoft, Active Directory, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Centrify software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,442,962 and 9,378,391.

The names of any other companies and products mentioned in this document may be the trademarks or registered trademarks of their respective owners. Unless otherwise noted, all of the names used as examples of companies, organizations, domain names, people and events herein are fictitious. No association with any real company, organization, domain name, person, or event is intended or should be inferred.

Contents

About this guide

This *Access Control and Privilege Management Scripting Guide* describes the Centrify Authentication & Privilege PowerShell-based command set. These PowerShell cmdlets run on Windows computers and can be used to automate access control and privilege management tasks, such as the creation of Centrify zones, rights, and roles. You can also use the cmdlets to perform other administrative tasks. For example, you can write scripts to add UNIX profiles for Active Directory users and groups to Centrify zones, assign UNIX and Windows users and groups to roles, and manage network information through NIS maps.

Intended audience

This guide provides information for Active Directory administrators who want to use PowerShell scripts to install or maintain Centrify software. This document supplements the help provided within the PowerShell environment using the `get-help` function. Whereas the `get-help` function describes each cmdlet in detail, this document provides an introduction to the Access Module for Windows PowerShell objects and how you can use PowerShell cmdlets and scripts to perform access control and privilege management tasks.

This guide assumes general knowledge of Microsoft Active Directory, of PowerShell scripts and syntax, and of the Windows PowerShell modules used to write scripts for Active Directory. You should also understand the structure of Active Directory, including the Active Directory schema your organization is using.

In addition to scripting skills, you should be familiar with Centrify architecture, terms, and concepts, and understand how to perform administrative tasks for authentication and privilege elevation and for the UNIX platforms you support.

Using this guide

This guide discusses access control and privilege management using PowerShell-based command-line programs. This information is intended to help you develop scripts for creating and populating zones and performing other administrative tasks on Windows computers. With scripts, you can automate the administrative tasks you might otherwise perform using the Access Manager console.

The guide provides the following information:

- Developing scripts for administrative tasks provides an introduction to access control and privilege management using Windows PowerShell.
- Installing the access module for PowerShell describes how to download and install the module as a separate package.
- Managing Centrify objects using Windows PowerShell scripts describes how to use the cmdlets to connect to Active Directory and perform access control and privilege management tasks.
- Objects and properties lists the objects defined by the authentication and privilege elevation PowerShell module, and the properties of each object.
- Adding users in a one-way trust environment explains how to add a user in a one-way trust environment by using the authentication and privilege elevation PowerShell module.
- Using predefined scripts to generate reports describes the predefined report scripts that are included with the authentication and privilege elevation PowerShell module, and how to configure report output files to generate HTML and PDF formatted report files.

Compatibility and limitations of this guide

The information in this guide is intended for use with Centrify Server Suite, version 5.1.x or later and Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service 2017.2 or later. Although intended to be accurate and up-to-date, interfaces are subject to change without notice and can become incompatible or obsolete when a newer version of the software is released.

In general, application programming interfaces are also intended to be backward-compatible, but are not guaranteed to work with older versions of

the software. Because the authentication and privilege elevation cmdlets are subject to change, enhancement, or replacement, the information in this guide can also become incomplete, obsolete, or unsupported in future versions. If you are unsure whether this guide is appropriate for the version of the software you have installed, you can consult the Centrify web site or Centrify Support to find out if another version of this guide is available.

If you are using a different version of Centrify Authentication Service, Privilege Elevation Service, and Audit & Monitoring Service, consult the Centrify Web site or Centrify Support to find out if another version of this guide is available. Because the authentication and privilege elevation cmdlets are subject to change, enhancement, or replacement, the information in this guide can also become incomplete, obsolete, or unsupported in future versions.

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

Previous Product Offering	Previous Product Offering	Description	Current Product Offering
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	
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	
	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.

Developing scripts for administrative tasks

The authentication and privilege elevation for Windows PowerShell consists of the following:

- Application programming interfaces in the form of PowerShell command-line programs, or cmdlets, that are packaged in dynamic link libraries (.DLLs).
- A PowerShell help file that includes complete cmdlet reference information and this scripting guide.
- Sample scripts to illustrate administrative tasks.
- Predefined scripts to generate reports.
- Individual help files for each predefined report script.

On Windows computers, you can use the authentication and privilege elevation module for Windows PowerShell to develop your own custom scripts that access, create, or modify Centrify-specific data in Active Directory.

Getting started with cmdlets for PowerShell

The Access Module for PowerShell consists of “cmdlets” that you can use to manage Centrify-specific information in Active Directory. A “cmdlet” is a lightweight command-line program that runs in the Windows PowerShell environment. In most cases, cmdlets perform a basic operation and return a Microsoft .NET Framework object to the next command in the pipeline.

The cmdlets in the Centrify module enable you to access, create, modify, and remove information about Centrify zones, including details about the user, group, and computer profiles defined in each zone; all aspects of the rights, roles, and role assignments applicable in each zone; and the available NIS

maps and NIS map entries for each zone. You can combine cmdlets and use them in scripts to automate administrative tasks, such as the provisioning of user and group profiles, or the creation of rights, roles, and role assignments.

In most cases, you can use cmdlets to manipulate Centrify objects in any type of zone. However, because the implementation of authorization differs greatly in hierarchical zones from authorization in classic zones, the Access Module for Window PowerShell cmdlets that enable you to create and work with rights, roles, or role assignments are only applicable in hierarchical zones. You should not use the cmdlets for rights, roles, and role assignments in classic zones.

Managing UNIX information from a Windows computer

You can use the cmdlets to work with information for any Centrify-managed computer and to manage UNIX profiles and access rights. However, you can only run the cmdlets on Windows-based computers that have the Windows PowerShell command-line shell available. If you want to develop scripts that run on UNIX computers, you can use the ADEdit program (`adedit`). The ADEdit application provides functionality similar to the cmdlets. For detailed information about using ADEdit, see the *ADEdit Command Reference and Scripting Guide*.

Writing programs in other languages

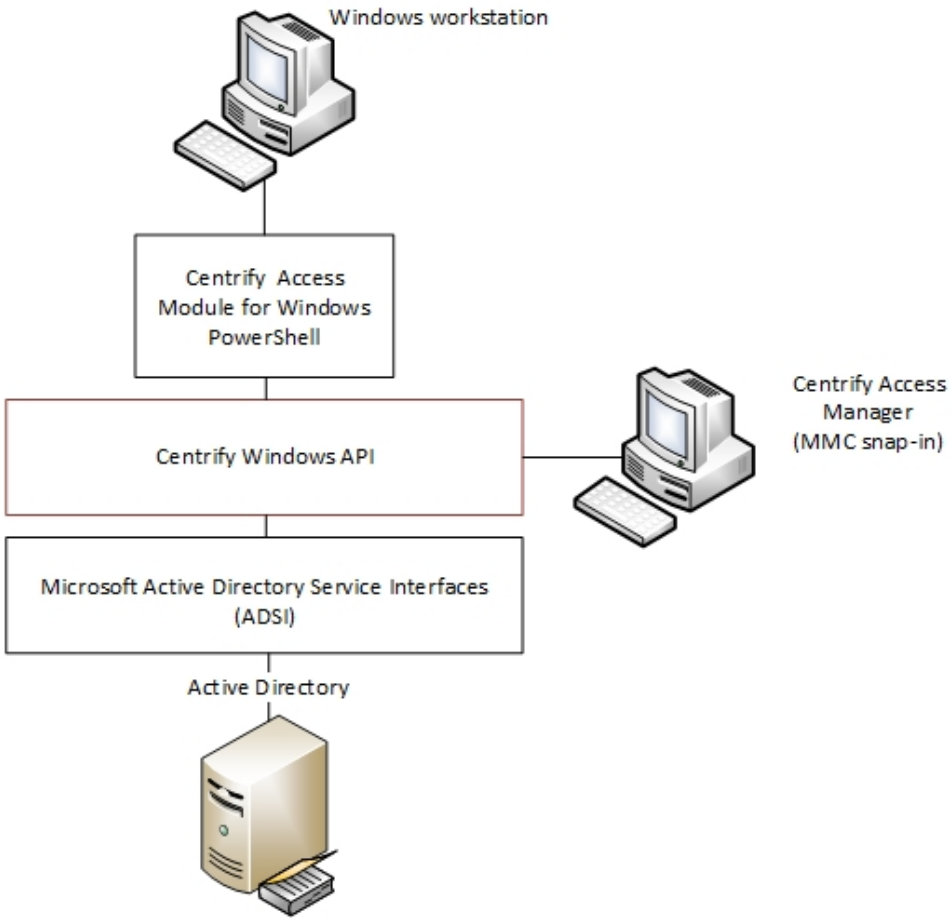
If you want to develop programs or scripts that run on Windows but outside of the Windows PowerShell environment, you can use any language that supports the Component Object Model (COM) interface. The Centrify COM-based interface is available as part of the Centrify Windows Software Development Kit (SDK). The SDK package is a completely separate application programming interface that provides reusable objects that you can call in programs written in .NET or COM-enabled languages. You can, therefore, create or modify your own applications to use these objects in VBScript and JScript or in .NET-compliant (C#) languages. For more information about using the COM-based API, see the *Windows API Programmer's Guide*.

Accessing information stored in Active Directory

The Centrify Access Module for PowerShell cmdlets connect to Active Directory to access all of the Centrify-specific information stored there. You can, therefore, write PowerShell scripts to automate procedures that you would otherwise have to perform using Access Manager.

The cmdlets rely on the underlying interfaces provided by Microsoft Active Directory Service Interfaces (ADSI) and the Centrify Windows API. The ADSI layer provides low-level functions that permit applications to read and write data in Active Directory. The cmdlets provide a task and object-based level of abstraction for retrieving and manipulating Centrify-specific information so that you do not need to know the details of how the data is stored or how to use any of the underlying ADSI functions directly.

The following figure illustrates how the Centrify Access Module for PowerShell provides a layer of abstraction between the data stored in Active Directory and your scripting environment.



The Active Directory schema defines how all of the objects and attributes in the database are stored. When you add Centrify objects to the Active Directory database, how that data is stored depends on the Active Directory schema you have installed. The Centrify Access Module for PowerShell, however, provides a logical view of the data, eliminating the need to know the details of how data is stored in different schemas when performing common administrative tasks. The cmdlets also provide a simple and Centrify-focused method for accessing UNIX objects that must be operated on.

Using the cmdlets, you can write scripts that automatically create and manage zones or update user, group, or computer properties. In most cases, the cmdlets enable you to perform exactly the same tasks from the command line that you would otherwise perform interactively using Access Manager.

Installing the access module for PowerShell

You can install the authentication and privilege elevation module for PowerShell from the authentication, privilege elevation, and audit and monitoring services setup program or as a separate package. It includes the access control and privilege management cmdlets for Windows PowerShell, sample scripts, and documentation for performing common administrative tasks using PowerShell scripts. This chapter describes how to install the software if you download it as a separate package or run the package-specific setup program on a Windows computer.

Selecting and downloading a standalone package

The cmdlets that run in Windows PowerShell are defined in dynamic link libraries that can be installed on any computer where you install other Windows-based components, such as the Access Manager console. You can also download these libraries separately, along with sample scripts and documentation, onto computers where Access Manager is not installed.

You can download the Access Module for PowerShell as a separate package from the Centrify Downoad Center under **Software Development Kits**. However, you must obtain an unlocking code or license key from your Centrify sales representative to access the module.

Running the setup program

After you have downloaded the compressed file to your computer, you can extract the files and run the setup program to install the Access Module for PowerShell files.

If you want to use the authentication and privilege elevation module for Windows PowerShell on a Server Core computer, however, you must have Windows PowerShell, version 2.0 or later, installed before attempting to install the module. Also, you need to install the the authentication and privilege elevation module for Windows PowerShell on a Windows Server Core environment in silent mode, due to a user interface limitation. Please check the process exit code to see whether the installation succeeded or failed. (Ref: CS-33696a)

To run the standalone setup program:

1. Select the downloaded file, right-click, then select **Extract All** to extract the compressed files to a folder.
2. Double-click the standalone executable to start the setup program interactively.

   For example, for the 64-bit version of the file, double click the `CentrifyDC_PowerShell-5.2.0-win64.exe` file.

   Alternatively, you can install from the Microsoft Installer (`.msi`) file. For example, you might run the following command:

   ```
   msiexec.exe /i "CentrifyDC_PowerShell-5.2.0-win64.msi"
   /norestart
   ```
3. At the Welcome page, click **Next**.
4. Select **I accept the terms in the License Agreement**, then click **Next**.
5. Accept the default location or click **Change** to choose a different location, then click **Next**.

   If you accept the default location the authentication and privilege elevation cmdlets are available in a separate authentication and privilege elevation for Windows PowerShell console.

   If you want the authentication and privilege elevation cmdlets to be available in the default Windows PowerShell console with other PowerShell modules, select the following location:

   ```
   C:\Windows\System32\WindowsPowerShell\v1.0\Modules\Centrify.DirectControl.PowerShell
   ```
6. Click **Install**.
7. Click **Finish** to complete the installation.

Importing the cmdlets into the Windows PowerShell console

If you install the authentication and privilege elevation module for Windows PowerShell in the default location, it is a self-contained Windows PowerShell console. If you install the files in the location for system modules so that cmdlets from other modules are available in the same console, you should import the authentication and privilege elevation module into your default Windows PowerShell console.

To import the authentication and privilege elevation module:

1. On the Start menu, select Windows PowerShell to display a menu extension with a list of Tasks.
2. On the Tasks menu, select Import System Modules to import the authentication and privilege elevation module and open the Windows PowerShell console.
3. Verify the installation and import completed successfully, type the following command:

   ```
   get-command *-Cdm*
   ```

You should see a listing of the authentication and privilege elevation cmdlets, similar to the following partial list:

```
PS C:\Windows\system32> get-command *-Cdm*
CommandType Name                           Definition
----------- ----                           ----------
Cmdlet      Add-CdmApplicationRight        Add-CdmApplicationRight -
Right ...
Cmdlet      Add-CdmCommandRight            Add-CdmCommandRight -Right
<Cdm...
Cmdlet      Add-CdmDesktopRight            Add-CdmDesktopRight -Right
<Cdm...
Cmdlet      Add-CdmNetworkAccessRight      Add-CdmNetworkAccessRight
-Righ...
Cmdlet      Add-CdmPamRight                Add-CdmPamRight -Right
<CdmPamR...
Cmdlet      Add-CdmSshRight                Add-CdmSshRight -Right
<CdmSshR...
Cmdlet     Get-CdmApplicationRight        Get-CdmApplicationRight [-
Zone ...
Cmdlet     Get-CdmCommandRight            Get-CdmCommandRight [-Zone
<Cdm...
Cmdlet     Get-CdmComputerRole            Get-CdmComputerRole -Zone
<CdmZ...
Cmdlet     Get-CdmDesktopRight            Get-CdmDesktopRight [-Zone
<Cdm...
```

```
Cmdlet     Get-CdmGroupProfile        Get-CdmGroupProfile [-Zone
<Cdm...
...
```

Managing Centrify objects using Windows PowerShell scripts

This chapter provides an overview of how you can use the cmdlets to access and manage authentication and privilege elevation information stored in Active Directory using Windows PowerShell scripts. It provides a summary of the operations you can perform using cmdlets and how to establish a connection to Active Directory. For more examples of how to perform common administrative tasks using the cmdlets in PowerShell scripts, see the samples included with the software.

Using cmdlets to manage access

The Centrify Access Module for PowerShell provides cmdlets that perform operations on objects that correspond to the core elements of Centrify data. The core elements of Centrify data for access control and privilege management are the following:

- Computers
- Users and user profiles
- Groups and group profiles
- Zones and zone properties
- UNIX and Windows rights
- User role definitions
- Computer role definitions

- Role assignments
- NIS network maps and map entries

In most cases, you can use cmdlets to manipulate Centrify information in any type of zone. However, because the implementation of authorization differs greatly in hierarchical zones from authorization in classic zones, the Access Module for Window PowerShell cmdlets that enable you to work with rights, roles, or role assignments are only applicable in hierarchical zones. You should not use the cmdlets for rights, roles, and role assignments in classic zones. Other than this limitation, you can use the cmdlets to create, access, modify, and remove information associated with any of the core elements of Centrify data for access control and privilege management.

Most of the cmdlets perform one of the following basic operations:

- `New-CdmXxx` cmdlets create new Centrify objects, such as a new zone or a new role definition.
- `Add-CdmXxx` cmdlets add a right to a specified role.
- `Get-CdmXxx` cmdlets get the properties of a specified object.
- `Set-CdmXxx` cmdlets set or change the properties of a specified object.
- `Remove-CdmXxx` cmdlets delete a specified object or remove a right from a specified role.

In addition to these basic operations, there are cmdlets for exporting and importing rights and roles from one zone to another and for establishing connections with Active Directory.

For reference information describing the use and parameters for each cmdlet, you can use the `get-help` function within the PowerShell console. For example, if you want to see a description and syntax summary for the `New-CdmZone` cmdlet, type the following command in the PowerShell console:

```
get-help New-CdmZone
```

If you want to see more detailed information about a cmdlet's parameters and code examples, you can use the `-detailed` or `-full` option. For example, type the following command in the PowerShell console:

```
get-help New-CdmZone -detailed
```

Creating and using a connection

Because the Centrify Access Module for PowerShell cmdlets manipulate objects in Active Directory, you must establish a connection with Active Directory before using cmdlets to perform other tasks. To establish a connection with Active Directory, you must specify a target domain or domain controller and the credentials to use when connecting to that domain or domain controller.

Once the credentials to use for connecting to a domain and the domain controller to use to connect to a domain are set, all subsequent calls share that information. You don't have to provide the credential or the domain controller for any subsequent calls.

The following example illustrates how to use the `administrator` account to connect to the `finance.acme` domain, then add the user `joe.doe` to the `Engineering` zone:

```
PS C:\> Set-CdmCredential "finance.acme" "administrator"
PS C:\> Get-CdmCredential
Target           Type    User
------           ----    ----
finance.acme     Forest  administrator@finance.acme
PS C:\> $zone = Get-CdmZone -Name "Engineering"
PS C:\> New-CdmUserProfile -Zone $zone -User
"joe.doe@finance.acme" -Login "jdoe"
```

In this example, the cmdlets that get the zone and create the user profile use the credential that is cached by `Set-CdmCredential` command. The `Get-CdmCredential` cmdlet shows what credentials are cached currently.

Managing connections

You can use the following cmdlets to manage connections to Active Directory by adding, modifying, or using cached credentials or specifying domain controller to domain mappings:

- Set-CdmCredential to add or modify a credential in the cache.
- Get-CdmCredential to list the credentials currently cached.
- Set-CdmPreferredServer to specify a domain controller to use for a domain.
- Get-CdmPreferredServer to list all domain controller to domain mapping previously defined.

Specifying credentials

You can use `Set-CdmCredential` cmdlet to specify a credential that you want to cache in the form of a `PSCredential` object. You can create the `PSCredential` object using the `Get-Credential` cmdlet. The `Get-Credential` cmdlet will prompt user interactively to specify a user name and password. You can also pass the user name as a parameter to the `Get-Credential` cmdlet to have the cmdlet prompt the user for the password.

If you want to specify the credentials to establish a connection with Active Directory without prompting for a password, you can hard code the user name and password for the `PSCredential` object into your script. For example:

```
$SecurePassword = "p@ssw0rd" | ConvertTo-SecureString -
AsPlainText -Force

$Credentials = New-Object
System.Management.Automation.PSCredential
  -ArgumentList "DOMAIN\user", $SecurePassword
```

In most cases, hard coding a password into a script is not a secure practice and is not recommended. However, it does allow you to write scripts that run without user interaction.

Organizing cmdlet operations in a sequence

There is no fixed sequence in which cmdlets must be called. There is, however, a logical sequence to follow to make information available from one to another. For example, to get all of the user UNIX profiles in a zone, you must first identify the zone object you want to work with before you call the `Get-CdmUserProfile` cmdlet, To accomplish this, you could organize the calls in the following sequence:

```
$zone = Get-CdmZone -Name "myZone"
Get-CdmUserProfile -Zone $zone
```

Similarly, to get all of the UNIX user profiles for a specific computer, you must first identify the computer object:

```
$computer = Get-CdmManagedComputer -Name "myComputer"
Get-CdmUserProfile -Computer $computer
```

In most cases, you can determine from the parameters of a cmdlet whether you need to call another cmdlet first. For example, if you want to add a right to a role, you must have created the role first so it can be specified as a parameter to the `Add-CdmXxx` cmdlet.

For most `Set-CdmXxx` or `Remove-CdmXxx` cmdlets, you must call the corresponding `Get-CdmXxx` or `Add-CdmXxx` cmdlet to obtain the object first. For example, to delete `"role1"` from `"zone1"`, you could call the cmdlets as follows:

```
Get-CdmRole -Zone "cn=zone1,cn=Zones,dc=acme,dc=com" -Name
"role1" | Remove-CdmRole
```

In this example, the `Get-CdmRole` cmdlet retrieves "`role1`" from the specified zone and passes it to the `Remove-CdmRole` cmdlet.

Checking for valid licenses

All of the authentication and privilege elevation cmdlets check for a valid license before performing the requested action. The license check succeeds only if one of the following conditions is true:

- There is at least one evaluation license that has not expired.
- There is at least one workstation license.
- There is at least one server license.

If the license check fails, the cmdlet displays an error and stops running. If the license check succeeds, the result is cached. The next time a cmdlet tries to access the same forest, it uses the cached result rather than performing the license check again. Note that the cache is only effective in one PowerShell console. If another PowerShell console runs a cmdlet accessing the same forest, the cmdlet in that console performs a separate license check.

Working with sample scripts

There are several sample scripts included with the software to demonstrate a few common administrative tasks. You can copy and modify these sample scripts to use them in your environment or study them as examples for writing your own custom scripts. The sample scripts include detailed comments about the operations performed to accomplish the following tasks.

This script	Illustrates this administrative task
backup.ps1	How to create a backup copy of a self-contained Centrify zone. This script creates an XML file that contains all computer, user, and group profiles, authorization information, and child zone information for a parent Centrify zone. You cannot use this script to backup SFU zones or child zones.
CreateZoneAndDelegate.ps1	How to create a new zone and delegate all zone administrative tasks to a specific trustee.
RemoveAllOrphans.ps1	How to find and delete all user, group, and computer profiles that no longer have a corresponding Active Directory account on all managed computers in each zone.
RemoveEmptyCompRoles.ps1	How to find and remove computer roles that have no members. This script is only applicable for hierarchical zones.
RemoveEmptyZones.ps1	How to find and remove zones that have no computers, users, or authorization information. This script will only remove a zone if it contains no user or group profiles, no joined computers, no role assignments, no computer roles, and no child zones. If any of these objects exist for a zone, the zone is not removed. This script is only applicable for hierarchical zones.
ResetOrphanChildZones.ps1	How to find child zones that no longer have a parent zone and reset them to be independent zones.
restore.ps1	How to restore a self-contained Centrify zone from a backup created using the backup.ps1 sample script.

To run a sample script:

1. Open the Centrify Access Module for PowerShell.
2. Verify you have permission to execute scripts.

   ```
   Get-ExecutionPolicy
   ```

   In most cases, the permission to execute scripts is restricted. You can use the `Set-ExecutionPolicy` to allow execution. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   For more information about execution policies and the options available, use the `get-help` function.
3. Verify you are in the directory where the scripts are located.

4. Execute the sample script.

   ```
   .\RemoveAllOrphans
   ```

Using the backup and restore scripts

If you want to use the sample backup and restore scripts to backup self-contained Centrify zones, you must modify the content of the scripts before executing them.

To run the sample backup script:

1. Open the `backup.ps1` file in a text editor.
2. Modify the path to the zone you want to back up and the path to the backup file at the start of the sample script.

   ```
   # Input the zone DN you want to backup
   $zoneDn = "CN=Headquarters,CN=Zones,OU=Centrify
   Pubs,DC=pistolas,DC=org"
   $xmlPath = "C:\Program Files\Centrify\HQ-test.xml"
   ```

3. Modify the confirmation message at the end of the script to display the path to the backup file.

   ```
   Write-Host "Backup to C:\Program Files\Centrify\HQ-test.xml
   is done."
   ```

4. Save your changes with a new file name—for example, `HQbackup.ps1`—to keep the sample `backup.ps1` script unchanged.
5. Open the Centrify Access Module for PowerShell.

   Alternatively, you can use the default Windows PowerShell console. If you use open the default Windows PowerShell console, run the import-module with the path to the Access Module for PowerShell libraries. For example, if you installed the module in the default location, run the following command to import the Centrify Access Module for PowerShell:

   ```
   import-module
   “C:\Program Files\Centrify\PowerShell\Centrify.DirectControl.
   PowerShell.dll”
   ```

6. Verify you have permission to execute scripts.

   ```
   Get-ExecutionPolicy
   ```

   In most cases, the permission to execute scripts is restricted. You can use the `Set-ExecutionPolicy` to allow execution. For example:

```
Set-ExecutionPolicy Unrestricted
```

For more information about execution policies and the options available, use the `get-help` function.

7. Verify you are in the directory where the scripts are located.
8. Execute the sample script.

```
.\HQbackup.ps1
```

To restore a zone from a backup file:

1. Open the `restore.ps1` file in a text editor.
2. Modify the path to the zone you want to restore and the path to the backup file at the start of the sample script.

```
## Input the zone container you want to create
$newZoneContainer = "CN=Zones,OU=Centrify
Pubs,DC=pistolas,DC=org
...
$xmlPath = "C:\Program Files\Centrify\HQ-test.xml"
```

3. Save your changes with a new file name—for example, `HQrestore.ps1`—to keep the sample `restore.ps1` script unchanged.
4. Open the Centrify Access Module for PowerShell.
5. Execute the sample script.

```
.\HQrestore.ps1
```

Creating new zones with the sample CreateZoneAndDelegate script

You can use the CreateZoneAndDelegate sample script to automate the creation of new zones and assign an Active Directory user or group to be the zone administrator. By default, the script delegates all administrative tasks to the user or group you specify. To use the script without modification, you simply need to specify the Active Directory container where you want to create the zone, the zone name, and the user or group who should be designated the zone administrator.

To create new zone using the sample script:

1. Open the Centrify Access Module for PowerShell.
2. Verify you are in the directory where the scripts are located.

3. Execute the sample script with the required command line arguments.

   ```
   .\CreateZoneAndDelegate -Container “cn=Zones,ou=Centrify
   Pubs,dc=pistolas,dc=org” -ZoneName seattle -trustee
   frank.smith@pistolas.org
   ```

4. Open Access Manager.
5. Select Zones, right-click, then select Open Zone to search for and select the new zone.

If you want to delegate specific administrative tasks, you can copy the sample script and modify the `Set-CdmDelegation` call to specify a list of tasks. For example:

```
Set-CdmDelegation -Zone $zone -Task "AddUsers",”AddGroups” -
Trustee $trustee;
Write-Host "$trustee is delegated the rights to add users and
groups.";
```

Generating reports from predefined scripts

Most of the predefined reports in Access Manager Report Center have a corresponding PowerShell script that you can use to generate reports from the PowerShell console. See Using predefined scripts to generate reports for details about generating reports based on predefined scripts.

Recommendations for writing custom scripts

Most cmdlets and scripts return information efficiently without any special handling or any noticeable effect on performance. If you plan to write custom scripts that could potentially return large data sets, however, you should consider ways to improve performance. For example, if you are writing a script that exports a large number of zones or reports on a large number of users, you might want to use the following recommendations as guidelines.

- When testing the performance of the script, use the standard `Measure-Command` cmdlet to accurately measure cmdlet and script performance.

  The `Measure-Command` cmdlet ignores the time it takes to print all of the results returned to the PowerShell console. In many cases, the execution of a script is efficient, but rendering the results in the PowerShell console might make the cmdlet or script performance seem unacceptable.

- Consider how you want to balance memory usage and performance when using the PowerShell pipeline if your cmdlet or script returns large data collections.

  For example, you might use `foreach` in a script instead of using the pipeline to improve performance.

  Use syntax similar to this:

  ```
  foreach ($cmd in Get-CdmUserProfile -Zone $z) { action_on_
  each_cmd }
  ```

  Instead of:

  ```
  Get-CdmUserProfile -Zone $z | action_on_each_cmd
  ```

  However, if you choose not to use the pipeline, you should keep in mind that all of the returned objects stay in memory and might cause an out-of-memory error. Therefore, you should try to maintain balance between the scripts memory usage and performance.

- Cache the data, if possible, by writing the results to a file.

  For example, to add 1000 users to a zone use syntax similar to this to get a zone once:

  ```
  $zone = Get-CdmZone -Dn "cn=QA,cn=Zones,dc=ajax,dc=org"
  $profile1 = New-CdmUserProfile -Zone $zone -User
  user1@ajax.org -Uid 10001
  ...
  $profile1000 = New-CdmUserProfile -Zone $zone -User
  user1000@ajax.org -Uid 11000
  ```

  Instead of using syntax like this, which gets the zone from its distinguished name (DN) for every user:

  ```
  $profile1 = New-CdmUserProfile -Zone
  "cn=QA,cn=Zones,dc=ajax,dc=org" -User user1@domain.com -Uid
  10001
  ...
  $profile1000 = New-CdmUserProfile -Zone
  "cn=QA,cn=Zones,dc=ajax,dc=org" -User user1000@domain.com -
  Uid 11000
  ```

- Use `Export-Csv` instead of `Out-File` if possible. The `Export-Csv` cmdlet writes results to a file faster than the `Out-File` cmdlet.

- If you are writing a script that generates a very large data set—for example, reporting information for a global zone—you might want to use the native .NET `FileStream` function. The `FileStream` function is the fastest way to write content to a file.

  For example, you might use a code snipper like this:

```
$fs = New-Object IO.FileStream <file>,
'Append','Write','Read'
    $fw = New-Object System.IO.StreamWriter $fs
        $zone = Get-CdmZone -Dn
"cn=global,cn=Zones,dc=ajax,dc=org"
       foreach ($cz in $zone) {$fw.WriteLine("{0} {1}",
$cz.Name, $cz.Type)}
   $fw.Close()
$fs.Dispose()
```

Enabling logging for cmdlets

For performance reasons, logging for cmdlets is disabled by default. To enable logging, you must modify the registry on the computer where you are running the Access Module for Windows PowerShell.

To enable logging:

1. Run `regedit` to open the Registry Editor
2. Select the HKEY_CURRENT_USER > Software > Centrify registry key.
3. Right-click, then select New > **Key** and type CIMS.
4. Select the new CIMS key, right-click, then select New > **String Value** with the name of `LogPath`.
5. Specify the path to the log file as the value.

   For example, set the value to `C: \Temp\Log`
6. Select the new CIMS key, right-click, then select New > **DWORD (32-bit) Value** with the name of `TraceLevel`.
7. Specify the level of detail to write to the log file as the value.

   The valid settings are:

   - 0 to disable logging.
   - 1 to only log error messages.
   - 2 to log errors and warning messages.
   - 3 to log errors, warnings, and informational messages.
   - 4 to log all debugging and tracing messages.

   For example, set the value to 4 to enable detailed logging of all messages.

Getting information about the cmdlets available

You can use the `get-help` command with different options to get summary about the cmdlets available in the Centrify Access Module for PowerShell or detailed information about the specific cmdlets you want to use. For example, you can use `get-help` with the `-full` command-line option to see complete reference information for a specified cmdlet or `get-help -example` to display only the examples for a specified cmdlet.

To see the current list of cmdlets available open the Centrify Access Module for PowerShell, then run the following command:

```
get-help *cdm*
```

This command displays a summary of the Centrify Access Module for PowerShell cmdlets similar to the following:

```
Name                                Category  Synopsis
----                                --------  --------
Add-CdmApplicationRight             Cmdlet    Adds a Windows
application right...
Add-CdmCommandRight                 Cmdlet    Adds a UNIX command
right to a s...
Add-CdmDesktopRight                 Cmdlet    Adds a Windows desktop
right to ...
Add-CdmNetworkAccessRight           Cmdlet    Adds a Windows network
access ri...
Add-CdmPamRight                     Cmdlet    Adds a PAM application
access ri...
Add-CdmSshRight                     Cmdlet    Adds an SSH application
right to...
Export-CdmData                      Cmdlet    Exports roles and
rights from th...
Get-CdmApplicationRight             Cmdlet    Gets an application
right from a...
Get-CdmCommandRight                 Cmdlet    Gets a command right
from a zone...
Get-CdmComputerRole                 Cmdlet    Gets a computer role
from a zone.
Get-CdmCredential                   Cmdlet    Gets user credentials.

Get-CdmDesktopRight                 Cmdlet    Gets a Windows desktop
right fro...
Get-CdmEffectiveGroupProfile        Cmdlet    Gets effective group
profiles fo...
Get-CdmEffectiveUnixRight           Cmdlet    Gets the effective UNIX
rights a...
Get-CdmEffectiveUserProfile         Cmdlet    Gets effective user
profiles for...
Get-CdmEffectiveWindowsRight        Cmdlet    Gets the effective
Windows right...
Get-CdmGroupProfile                 Cmdlet    Gets group UNIX
profiles.
Get-CdmManagedComputer              Cmdlet    Gets zoned or auto-
```

```
zoned managed...
Get-CdmNetworkAccessRight       Cmdlet    Gets a Windows network
applicati...
Get-CdmNisMap                   Cmdlet    Gets NIS maps for the
specified ...
Get-CdmNisMapEntry              Cmdlet    Gets NIS map entries
for the spe...
Get-CdmPamRight                 Cmdlet    Gets a PAM application
access ri...
Get-CdmPreferredServer          Cmdlet    Gets domain to server
mapping.
Get-CdmRole                     Cmdlet    Gets roles from a zone.

Get-CdmRoleAssignment           Cmdlet    Gets role assignments.

Get-CdmSshRight                 Cmdlet    Gets an SSH application
right fr...
Get-CdmUserProfile              Cmdlet    Gets user UNIX
profiles.
Get-CdmZone                     Cmdlet    Gets the zone object.

Import-CdmData                  Cmdlet    Imports roles and
rights into a ...
New-CdmApplicationRight         Cmdlet    Creates a new Windows
applicatio...
New-CdmCommandRight             Cmdlet    Creates a new command
right in a...
New-CdmComputerRole             Cmdlet    Creates a new computer
role in a...
New-CdmDesktopRight             Cmdlet    Creates a new Windows
desktop ri...
New-CdmGroupProfile             Cmdlet    Creates a new UNIX
group profile.
New-CdmManagedComputer          Cmdlet    Pre-creates a computer
or comput...
New-CdmMatchCriteria            Cmdlet    Creates a new match
criteria for...
New-CdmNetworkAccessRight       Cmdlet    Creates a new Windows
network ac...
New-CdmNisMap                   Cmdlet    Creates a new NIS map
in a speci...
New-CdmNisMapEntry              Cmdlet    Creates a new NIS map
entry in a...
New-CdmPamRight                 Cmdlet    Creates a new PAM
application ac...
New-CdmRole                     Cmdlet    Creates a new role in a
zone.
New-CdmRoleAssignment           Cmdlet    Creates a new role
assignment.
New-CdmUserProfile              Cmdlet    Creates a new UNIX user
profile.
New-CdmZone                     Cmdlet    Creates a new zone.

Remove-CdmApplicationRight      Cmdlet    Deletes a Windows
application ri...
Remove-CdmCommandRight          Cmdlet    Deletes a command right
or remov...
```

• • • • • •

Remove-CdmComputerRole	Cmdlet	Deletes a computer role from a z...
Remove-CdmDesktopRight	Cmdlet	Deletes a Windows desktop right ...
Remove-CdmGroupProfile	Cmdlet	Deletes a UNIX group profile.
Remove-CdmManagedComputer	Cmdlet	Removes a managed computer from ...
Remove-CdmNetworkAccessRight	Cmdlet	Deletes a Windows network access...
Remove-CdmNisMap	Cmdlet	Deletes a NIS map from a zone.
Remove-CdmNisMapEntry	Cmdlet	Deletes a map entry from a NIS map.
Remove-CdmPamRight	Cmdlet	Deletes a PAM application access...
Remove-CdmRole	Cmdlet	Deletes a role.
Remove-CdmRoleAssignment	Cmdlet	Deletes a role assignment from a...
Remove-CdmSshRight	Cmdlet	Removes an SSH right from a role.
Remove-CdmUserProfile	Cmdlet	Deletes a UNIX user profile.
Remove-CdmZone	Cmdlet	Deletes an existing zone.
Set-CdmApplicationRight	Cmdlet	Updates an existing Windows appl...
Set-CdmCommandRight	Cmdlet	Updates an existing command right.
Set-CdmComputerRole	Cmdlet	Updates an existing computer role.
Set-CdmCredential	Cmdlet	Adds a user credential.
Set-CdmDelegation	Cmdlet	Updates the delegation of admini...
Set-CdmDesktopRight	Cmdlet	Updates an existing Windows desk...
Set-CdmGroupProfile	Cmdlet	Updates an existing UNIX group p...
Set-CdmNetworkAccessRight	Cmdlet	Updates an existing Windows netw...
Set-CdmNisMap	Cmdlet	Updates an existing NIS map.
Set-CdmNisMapEntry	Cmdlet	Updates an existing NIS map entry.
Set-CdmPamRight	Cmdlet	Updates an existing PAM applicat...
Set-CdmPreferredServer	Cmdlet	Specifies a preferred server.
Set-CdmRole	Cmdlet	Updates an existing role.
Set-CdmRoleAssignment	Cmdlet	Updates an existing role assignm...
Set-CdmUserProfile	Cmdlet	Updates an existing UNIX user pr...
Set-CdmZone	Cmdlet	Updates an existing

- - - - - -

zone.

Objects and properties

This chapter provides an alphabetical listing of the objects and the properties of each object defined in the Access module for PowerShell. Note that not all properties are available as parameters in the PowerShell cmdlets.

CdmAdObject

Represents an Active Directory object. The following properties are defined for this object.

Property	Type	Description
`Class`	`string`	Class of the Active Directory object.
`DistinguishedName`	`string`	Distinguished name of the Active Directory object.
`Guid`	`Guid`	Globally unique identifier (GUID) of the Active Directory object.
`Name`	`string`	Name of the Active Directory object.

CdmAdPrincipal

Represents an Active Directory account principal. The following properties are defined for this object.

Property	Type	Description
`Class`	`string`	Class of the Active Directory object.
`DistinguishedName`	`string`	Distinguished name of the Active Directory object.
`Guid`	`Guid`	Globally unique identifier (GUID) of the Active Directory object.
`Name`	`string`	Name of the Active Directory object.

Property	Type	Description
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	Security identifier (SID) of the Active Directory principal.

CdmApplicationRight

Represents a Windows application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
Description	string	Description of the application right.
IsRequireMfa	Boolean	Indicates whether the application right requires multi-factor authentication.
MatchCriteria	MatchCriteria[]	Filter criteria defined by an array of MatchCriteria objects that identifies the application associated with the application right.
Name	string	Name of the application right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority of the application right; highest priority prevails.
RequirePassword	Boolean	Indicates whether the application right requires authentication.
RunasSelfGroups	Group	The group privileges to add to the user's account when running the application associated with the application right.
RunasUser	User	The user to run the application as.
Zone	Zone	Zone where the application right is defined.

CdmCommandRight

Represents a UNIX command right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
AddVar	string	Comma separated list of environment variable name-value pairs to add to the final list resulting from KeepVar or DeleteVar property (e.g. "var1=a,var2=b,var3=c").
Authentication	string	The authentication type of the command right: none, user, or runastarget.
DeleteVar	string	Comma separated list of environment variables to remove from default set when command is run.
Description	string	Description of the command right.
Digests	string	Specifies SHA-2 digests so that sudo can verify the binary's checksum (SHA-2) before sudo executes the binary. The supported hash types are as follows: ■ sha224 ■ sha256 ■ sha384 ■ sha512
DzdoRunAsGroup	string	Comma-separated string of groups allowed to run this command using dzdo (for example, "group1,group2,group3"). ■ The asterisk wild card (*) means any group enabled for the zone can run the command. ■ An empty string ("") means the command cannot run as any group.
DzdoRunAsUser	string	Comma-separated list of users allowed to run this command using dzdo (for example, "user1,user2,user3"). ■ The asterisk wild card (*) means any user enabled for the zone can run the command. ■ An empty string ("") means the command cannot run as any user.
DzshRunas	string	The user this command will run as under dzsh, '$' means current user.
IsAllowNested	Boolean	True if the command is allowed to start another program or open a new shell.
IsDisablePathTraverse	Boolean	True if the command does not allow navigation up the path hierarchy as an argument.
IsPreserveGroup	Boolean	True to retain the user's group membership while executing a command.

Property	Type	Description
IsRequireMfa	Boolean	Indicates whether the command right requires multi-factor authentication.
KeepVar	string	Comma separated list of environment variables to keep in addition to those in dzdo.env_keep when command is run.
MatchPath	string	The path for matching the command.
Name	string	Name of the command right.
Pattern	string	Command pattern for matching the command.
PatternType	string	The type of pattern—`glob` or `regexp`—used to match the command.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority for this command; highest priority prevails.
SELinuxRole	string	Sets the SELinux security context to use the specified role when executing a command using dzdo or dzsh. Applies to command rights on Red Hat Enterprise Linux systems that have SELinux enabled and are joined to a hierarchical zone.
SELinuxType	string	Sets the SELinux security context to use the specified type when executing a command using dzdo or dzsh. Applies to command rights on Red Hat Enterprise Linux systems that have SELinux enabled and are joined to a hierarchical zone.
UMask	string	User file-creation mode mask (`umask`) value that defines who can execute the command.
Zone	CdmZone	Zone of the command right.

CdmComputer

Represents an Active Directory computer object. The following properties are defined for this object.

Property	Type	Description
Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
DNSHostName	string	DNS host name of the Active Directory computer.

Property	Type	Description
Enabled	Boolean	True if the Active Directory computer is enabled.
Guid	Guid	GUID of the Active Directory object.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal.
UserPrincipalName	string	User principal name of the Active Directory computer.

CdmComputerRole

Represents a Centrify computer role. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
CustomAttributes	string	Custom text strings for the computer role.
Description	string	Description of the computer role.
Group	CdmGroup	Computer group associated with this computer role.
Name	string	Name of the computer role.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone that contains the computer role.

CdmDesktopRight

Represents a Windows desktop access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
Description	string	Description of the desktop right.
IsRequireMfa	Boolean	Indicates whether the desktop right requires multi-factor authentication.
Name	string	Name of the desktop right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.

Property	Type	Description
Priority	int	Priority of the desktop right; highest priority prevails.
RequirePassword	Boolean	True if the desktop right requires a password.
RunasSelfGroups	CdmGroup []	Groups whose privileges are added to the user account running the desktop.
RunasUser	CdmUser	User to run the desktop as.
Zone	CdmZone	Zone of the desktop right.

CdmEffectiveUnixRights

Represents the UNIX rights assigned to a user that are in effect on a Linux or UNIX computer in a zone. The following properties are defined for this object.

Property	Type	Description
AuditLevel	string	Effective auditing level.
CommandRights	CdmEffectiveCommandRight []	The array of effective command rights assigned to the user.
Computer	CdmManagedComputer	The computer in which the roles and rights are effective.
HasRescueRight	Boolean	True if the user has the rescue right.
PamRights	CdmEffectivePamRight[]	The array of effective PAM rights assigned to the user.
Profiles	CdmEffectiveUserProfile []	Effective UNIX profiles for the Active Directory user in the computer.
Roles	CdmEffectiveRole[]	The array of effective roles assigned to the user.
SshRights	CdmEffectiveSshRight[]	The array of effective SSH rights assigned to the user.
UnixSystemRights	string[]	Effective UNIX system rights.
User	CdmUser	Active Directory user assigned to the role.

CdmEffectiveWindowsRights

Represents the Windows rights assigned to a user that are in effect on a Windows computer in a zone. The following properties are defined for this object.

Property	Type	Description
AuditLevel	string	Effective auditing level.
ApplicantioRights	CdmEffectiveApplicationRight	The array of effective application rights assigned to the user.
Computer	CdmManagedComputer	The computer in which the roles and rights are effective.
DesktopRights	CdmEffectiveDesktopRight	The array of effective desktop rights assigned to the user.
HasRescueRight	Boolean	True if the user has the rescue right.
NetworkRights	CdmEffectiveNetworkRigh	The array of effective network access rights assigned to the user.
Roles	CdmEffectiveRole	The array of effective roles assigned to the user.
WindowsSystemRights	string[]	Effective Windows system rights.
User	CdmUser	Active Directory user assigned to the role.

CdmGroup

Represents an Active Directory group. The following properties are defined for this object.

Property	Type	Description
Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
GroupCategory	ADGroupCategory	Category of the Active Directory group.
GroupScope	ADGroupScope	Scope of the Active Directory group.
Guid	Guid	GUID of the Active Directory object.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal.

CdmGroupProfile

Represents a UNIX group profile. The following properties are defined for this object.

Property	Type	Description
Computer	CdmManagedComputer	Computer that contains the profile.
Gid	long	GID of the group profile.
Group	CdmGroup	Active Directory group of the group profile.
IsHierarchical	Boolean	True if the group profile is in a hierarchical zone.
IsMembershipRequired	Boolean	True if users are required to be a member of this group.
IsOrphan	Boolean	True if the group profile is an orphan profile, that is, it has no corresponding Active Directory group.
IsSfu	Boolean	True if the group profile is a SFU profile.
Name	string	Name of the group profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone that contains the profile.

CdmLocalGroupProfile

Represents a local UNIX group profile. The following properties are defined for this object.

Property	Type	Description
CanonicalName	string	Canonical name of the local group profile.
Computer	CdmManagedComputer	Computer where the local group profile is defined.
Domain	string	Domain of the local group profile.
Gid	long	GID of the group profile.
Members	string[]	Members of the local group profile.
Name	string	Name of the group profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.

Property	Type	Description
State	enum	State of the local group profile. The valid values are: ■ Enable ■ Remove ■ Inherit The default state is Inherit.
Zone	CdmZone	Zone that contains the profile.

CdmLocalUserProfile

Represents a local UNIX user profile. The following properties are defined for this object.

Property	Type	Description
CanonicalName	string	Canonical name of the local user profile.
Computer	CdmManagedComputer	Computer where the local user profile is defined.
Domain	string	Domain of the local user profile.
Gecos	string	GECOS field of the local user profile.
HomeDir	string	Home directory of the user associated with the local profile.
Name	string	Name of the user associated with the local profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
PrimaryGroupId	long	Primary group ID of the user associated with the local profile.
Shell	string	Default shell of the user associated with the local profile.
State	enum	State of the local user profile. The valid values are: ■ Enable ■ Remove ■ Inherit The default state is Inherit.

Property	Type	Description
Uid	long	Numeric user identifier (UID) of the user associated with the local profile.
Zone	CdmZone	Zone where the local user profile is defined.

CdmLocalWindowsGroup

Represents a local Windows group account. The following properties are defined for this object.

Property	Type	Description
CanonicalName	string	Canonical name of the local group in Active Directory.
Computer	CdmManagedComputer	Computer where the local group is defined.
Description	string	Description for the local group.
Domain	string	Domain of the local group in Active Directory.
Members	string[]	Members of the local group .
Name	string	Name of the local group .
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
State	LocalWindowsGroupState enum	State of the local group . The valid values are: ■ Inherit ■ Enable ■ Remove The default state is Inherit.
Zone	CdmZone	The zone where the local group is defined.

CdmLocalWindowsUser

Represents a local Windows user account. The following properties are defined for this object.

Property	Type	Description
CanonicalName	string	Canonical name of the local user account in Active Directory.
Computer	CdmManagedComputer	Computer where the local user is defined.
Description	string	Description for the local user.
Domain	string	Domain of the local user account in Active Directory.
FullName	string	Full name of the local user.
Name	string	Name of the local user.
PasswordOptions	LocalWindowsUserPassword Option enum	Password options of the local user. Possible values are: ■ None ■ Inherit ■ UserMustChangePasswordAtNextLogon ■ UserCannotChangePassword ■ PasswordNeverExpires Remarks: It can be a combination of UserMustChangePasswordAtNextLogon and PasswordNeverExpires, UserCannotChangePassword and PasswordNeveExpires.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
State	LocalWindowsUserState enum	State of the local user. The valid values are: ■ Inherit ■ Enable ■ Disable ■ Remove The default state is Inherit.
Zone	CdmZone	The zone where the local user is defined.

CdmManagedComputer

Represents a computer managed by authentication and privilege elevation. The following properties are defined for this object.

Property	Type	Description
AgentVersion	string	Version number of the Centrify agent installed on the managed computer.
Computer	CdmComputer	Corresponding Active Directory computer account.
ComputerZonePath	string	Path to the computer zone.
IsComputerZoneOnly	Boolean	True if the managed computer has a computer zone only (that is, the computer is not joined to a zone).
IsExpressMode	Boolean	True if the managed computer is in Express (unlicensed) mode.
IsHierarchical	Boolean	True if the managed computer is joined to a hierarchical zone.
IsOrphan	Boolean	True if the managed computer is an orphan profile, that is, it has no corresponding Active Directory computer object.
IsWindows	Boolean	True if the managed computer is a Windows computer.
IsWorkstationMode	Boolean	True if the managed computer is joined to Auto Zone in Workstation mode.
IsJoinedToZone	Boolean	True if the managed computer is joined to a zone.
LicenseType	string	Type of license being used. This property is Server if the managed computer is a Windows or UNIX server or Workstation if the managed computer is not used as a server.
Name	string	Name of the managed computer.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
ScpPath	string	Path to the service connection point for the managed computer.
Zone	CdmZone	Zone of the managed computer.

CdmMatchCriteria

Represents an application right match criteria object defined using the application rights match criteria filters. The following properties are defined for this object.

Property	Type	Description
FileType	string	The file type for an application.
FileName	string	The file name for an application.

Property	Type	Description
Path	string	The path to an application.
Argument	string	The argument for the application.
IsArgumentCaseSensitive	Boolean	True if the argument specified is case sensitive.
IsArgumentExactMatch	Boolean	True if the argument must be matched exactly as specified.
ProductName	string	All or part of the product name associated with the application.
ProductNameMatchOption	string	Specifies whether the product name string should be an exact match (is) or a partial match (contains).
CompanyName	string	All or part of the company name associated with the application.
CompanyNameMatchOption	string	Specifies whether the company name string should be an exact match (is) or a partial match (contains).
FileDescription	string	All or part of the file description for the application.
FileDescriptionMatchOption	string	Specifies whether the file description string should be an exact match (is) or a partial match (contains).
LocalOwnerType	string	The local owner type for the application.
LocalOwner	string	The local owner for the application.
OwnerSid	string	The owner security identifier (SID) for the application.
ProductVersion	string	All or part of the product version information for an application.
ProductVersionMatchOption	string	Specifies whether the product version string should be an exact match (equal), an earlier or equal version (earlier or equal), or a later or equal version (later or equal).
FileVersion	string	All or part of the file version information for an application.
FileVersionMatchOption	string	Specifies whether the file version string should be an exact match (equal), an earlier or equal version (earlier or equal), or a later or equal version (later or equal).
FileHash	string	The file hash for an application.
Publisher	string	The publisher for an application.

Property	Type	Description
PublisherMatchOption	string	Specifies whether the publisher string should be an exact match (is), a partial match (contains), start with, or end with the specified string.
SerialNumber	string	The serial number for an application.
SerialNumberMatchOption	string	Specifies whether the serial number string should be an exact match (is), a partial match (contains), start with, or end with the specified string.
IsRequireAdministrator	Boolean	True if the application requires administrator privileges to execute.
Description	string	The description for the application criteria.

CdmNetworkRight

Represents a Windows network access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
Description	string	Description of the network right.
IsRequireMfa	Boolean	Indicates whether the network access right requires multi-factor authentication.
Name	string	Name of the network right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Priority	int	Priority of the network right; highest priority prevails.
RequirePassword	Boolean	True if the network right requires a password.
RunasSelfGroups	CdmGroup []	Groups whose privileges are added to the user account accessing the network.
RunasUser	CdmUser	Run-as user of the network right.
Zone	CdmZone	Zone of the network right.

CdmPamRight

Represents a PAM application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
Application	string	PAM application for this right.
Description	string	Description of the PAM access right.
Name	string	Name of the PAM access right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone of the PAM access right.

CdmRole

Represents a authentication and privilege elevation role. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
AllowLocalUser	Boolean	True if the role can be assigned to a local user.
AuditLevel	string	Audit setting for this role.
CustomAttributes	string	Custom text strings for the role.
Description	string	Description of the role.
HasRescueRight	Boolean	True if this role can operate without being audited in case of audit system failure.
Name	string	Name of the role.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
TimeBox	Hashtable	Active time of the role.
RequireMfa	Boolean	True if the role requires multi-factor authentication.
UnixSystemRights	string[]	UNIX system rights granted to the role.
WindowsSystemRights	string[]	Windows system rights granted to the role.
Zone	CdmZone	Containing zone.

CdmRoleAssignment

Represents a authentication and privilege elevation role assignment. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
AdTrustee	CdmAdPrincipal	The trustee, if it is an Active Directory account.
Computer	CdmManagedComputer	Containing computer.
ComputerRole	CdmComputerRole	Containing computer role.
CustomAttributes	string	Custom text strings for the role assignment.
Description	string	The role assignment description.
EndTime	DateTime	The ending date and time for the role assignment.
IsNeverExpire	Boolean	True if the role assignment never expires.
IsRoleOrphaned	Boolean	True if the role is missing or invalid.
IsStartImmediately	Boolean	True if the role assignment starts immediately.
IsTrusteeOrphaned	Boolean	True if the trustee is missing or invalid.
LocalTrustee	string	The trustee, if it is a local account.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Role	CdmRole	Assigned role.
StartTime	DateTime	The starting date and time for the role assignment.
TrusteeType	string	Type of trustee.
Zone	CdmZone	Containing zone.

CdmSshRight

Represents an SSH application access right. This object is only applicable in hierarchical zones. The following properties are defined for this object.

Property	Type	Description
Application	string	Secure shell application for this right.
Description	string	Description of the SSH right.
Name	string	Name of the SSH right.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
Zone	CdmZone	Zone of the SSH right.

CdmUser

Represents an Active Directory user. The following properties are defined for this object.

Property	Type	Description
Class	string	Class of the Active Directory object.
DistinguishedName	string	Distinguished name of the Active Directory object.
Enabled	Boolean	True if the Active Directory user is enabled.
GivenName	string	Given name of the Active Directory user.
Guid	Guid	GUID of the Active Directory object.
IsAllADUser	Boolean	True if the user is an Active Directory domain user account.
Name	string	Name of the Active Directory object.
SamAccountName	string	SAM account name of the Active Directory principal.
Sid	SecurityIdentifier	SID of the Active Directory principal
Surname	string	Surname of the Active Directory user.
UserPrincipalName	string	User principal name of the Active Directory user.

CdmUserProfile

Represents a UNIX user profile. The following properties are defined for this object.

Property	Type	Description
Computer	CdmManagedComputer	Containing computer.
Gecos	string	GECOS field of the user profile.
HomeDirectory	string	Home directory of the user associated with the profile.
IsHierarchical	Boolean	True if the user profile is in a hierarchical zone.
IsOrphan	Boolean	True if the user profile is an orphan profile, that is, it has no corresponding Active Directory user.
IsSecondary	Boolean	True if the user profile is a secondary profile.

Property	Type	Description
IsSfu	Boolean	True if the user profile is an SFU profile.
IsUseAutoPrivateGroup	Boolean	True if the user private group is to be used as the primary group.
Name	string	Name of the user associated with the profile.
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
PrimaryGroupId	long	Primary group ID of the user associated with the profile.
Shell	string	Default shell of the user associated with the profile.
Uid	long	UID of the user associated with the profile.
UnixEnabled	Boolean	True if the user profile is enabled for a classic zone. This property is not applicable in hierarchical zones.
User	CdmUser	Active Directory user for whom this is the user profile.
Zone	CdmZone	Containing zone.

CdmZone

Represents a Centrify zone. The following properties are defined for this object.

Property	Type	Description
AgentlessPasswordAttribute	string	Attribute in which to store the password hash for agentless client.
AvailableShells	string[]	Array of available shells that can be used as the default shell for zone users.
CanonicalName	string	Canonical name of the zone.
CloudInstance	String	Cloud instance URL to which the zone connects.
DefaultGecos	string	Default GECOS field for zone users.
DefaultGid	long	Default GID value to use for zone groups.
DefaultGroupName	string	Default group name to use for zone groups.
DefaultHomeDirectory	string	Default home directory for zone users.
DefaultPrimaryGroup	string	Default primary group to use for zone users.

Property	Type	Description
DefaultShell	string	Default shell for zone users.
DefaultUid	long	Default UID value to use for zone users.
DefaultUserName	string	Default user name to use for zone users.
DefaultValueZone	CdmZone	Zone to use as the source for default values in a selected zone.
Description	string	Description of the zone.
DistinguishedName	string	Distinguished name of the zone.
Domain	string	Active Directory domain associated with the zone.
IsBlockGroupInheritance	Boolean	True if groups defined in a parent zone are not inherited, and therefore not visible, in a child zone. This property is only applicable for hierarchical zones.
IsHierarchical	Boolean	True if it is a hierarchical zone.
IsOrphanChildZone	Boolean	True if the zone is a child zone with no parent zone (Hierarchical zone only).
IsSfu	Boolean	True if it is a SFU zone.
Name	string	Name of the zone.
NextGid	long	Next GID value available for assignment to a zone group.
NextUid	long	Next UID value available for assignment to a zone user.
NisDomain	string	NIS domain for SFU zone or agentless mode.
Parent	CdmZone	Parent zone (Hierarchical zone only).
PreferredServer	string	Preferred server to use for committing changes to Active Directory.
ReservedGid	long	Reserved GID values that cannot be assigned to a zone group.
ReservedUid	long	Reserved UID values that cannot be assigned to a zone user.
Schema	string	Schema of the zone.
SfuDomain	string	SFU domain of the zone (SFU zone only).
TenantId	String	The TenantId of the zone
TruncateUserName	Boolean	True if user names longer than 8 characters are automatically truncated for the zone.
Type	string	Type of the zone.
Variables	string []	Array of runtime variables.

Adding users in a one-way trust environment

Some operations, such as adding a user to a zone, may require more than one credential. For example, if you want to add a user from one forest to a zone in another forest when there is a one-way trust between the forest, you might need to specify credentials for each forest. This appendix explains how to add a user in a one-way trust environment when using PowerShell cmdlets.

Using a single account credential

If you want to add the user `targetuser`, who has a domain user account in `forest2.net` to the `zone1` in `forest1.net`, where `forest1.net` trusts `forest2.net` (a one-way trust), you must use an account that has the following permissions:

- Permission to add a user to `zone1` in `forest1.net`.
- Permission to read accounts in `forest2.net`.

If you have a single account with the appropriate permissions—for example, `superuser` in `forest2.net`—you can add the `targetuser` from `forest2.net` to the `zone1` in `forest1.net` as follows:

```
Set-CdmCredential "forest1.net" "forest2\superuser"
New-CdmUserProfile -Zone "cn=zone1,cn=Zones,dc=forest1,dc=net"
    -User "cn=targetuser,cn=Users,dc=forest2,dc=net"
    -login "UNIXname" -uid nnnnn
```

where UNIXname is the UNIX login name of `targetuser` and nnnn is the UID of the `targetuser`.

Using two account credentials

If you don't have a single account with the appropriate permissions in the two forests, adding the `targetuser` to a zone in another forest will require two accounts credentials. For example, you must identify accounts with the following permissions:

- An account in `forest1.net`that has permission to add a user to `zone1` (`user1`).
- An account in `forest2.net` that has read permission on `forest2.net` (`user2`).

After you identify the accounts with the appropriate permissions—for example, `user1` in `forest1.net` and `user2` in `forest2.net`—you can add the `targetuser` from `forest2.net` to the `zone1` in `forest1.net` as follows:

```
Set-CdmCredential "forest1.net" "forest1\user1"
Set-CdmCredential "forest2.net" "forest2\user2"
New-CdmUserProfile `
    -Zone "cn=zone1,cn=Zones,dc=forest1,dc=net" `
    -User "targetUser@forest2.net" `
       -login "UNIXname" `
       -uid nnnnn
```

where UNIXname is the UNIX login name of `targetuser` and nnnn is the user's UID.

Using predefined scripts to generate reports

Most of the predefined reports in Access Manager Report Center have a corresponding PowerShell script that you can use to generate reports from the PowerShell console. When you use a PowerShell script to generate a report, the report content displays as text in the PowerShell console window. You can optionally format the report content as an HTML or PDF file using third-party tools.

Provided report scripts

The following report scripts are included with authentication and privilege elevation PowerShell. The scripts are typically installed in the following folder:

```
C:\Program Files\Centrify\PowerShell\Centrify.DirectControl.PowerShell\Reports
```

For details about script syntax, parameters, and examples, see the script help files. Execute the PowerShell `Get-Help` command to display the help for a script. For example, to display help details for the `ZonesReport.ps1` script, execute the following command from the PowerShell command line:

```
PS> Get-Help .\ZonesReport.ps1 -Detailed
```

This script	Reports this information	Is equivalent to this Access Manager Report Center report
AuthorizationReportForComputers.ps1	Lists each computer in the zone and indicates which users are allowed to access each computer. This report applies to classic zones only. This report includes details from the user's UNIX profile for each user listed, including the user's Active Directory user name, UNIX user name, zone, UID, shell, home directory and primary group.	Classic Zone – Authorization Report for Computers
AuthorizationReportForUsers.ps1	Lists each user account in the zone and indicates which computers each user can access. This report applies to classic zones only. This report includes details from the user's UNIX profile for each user listed, including the user's UNIX user name, zone, UID, shell, home directory and primary group.	Classic Zone – Authorization Report for Users
ComputerEffectiveAuditLevelReport.ps1	Lists the audit level in effect for all authorized users on computers in each zone. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Audit Level
ComputerEffectiveRightsReport.ps1	Lists the privileges granted on each computer. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Rights
ComputerEffectiveRolesReport.ps1	Lists the roles assigned on each computer. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Effective Roles
ComputerRoleAssignmentsReport.ps1	Lists the computer roles that are defined for each zone. The report includes the users and groups and their associated roles. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Role Assignments

This script	Reports this information	Is equivalent to this Access Manager Report Center report
ComputerRoleMembershipReport.ps1	Lists the computer roles that are defined for each computer and the zone to which they belong. This report applies to hierarchical zones only.	Hierarchical Zone – Computer Role Membership Report
ComputersReport.ps1	Lists computer account information for each computer in each zone. The information displayed includes the computer account name in Active Directory, the computer's DNS name, the computer's operating system, and the version of the Centrify UNIX agent installed on the computer, if available.	Computers Report
GroupsReport.ps1	Lists group information for each group in each zone. The information that is displayed includes the Active Directory group name, the UNIX group name, the UNIX group identifier (GID), and whether the group is an orphan.	Groups Report
StaleComputersReport.ps1	Lists information about all authentication service-enabled computers that have not changed their password in a specified number of days (90 days by default).	Stale Computers Report
UnixUserEffectiveRightsReport.ps1	Lists the effective rights for each UNIX user on each computer. The report shows the name of the right, its type, and where it is defined. This report applies to hierarchical zones only.	Hierarchical Zone – UNIX User Effective Rights

This script	Reports this information	Is equivalent to this Access Manager Report Center report
UserAccountReport.ps1	Lists Active Directory account details for the users that have UNIX profiles in each zone. The report includes the Active Directory display name; the Active Directory logon name; the Active Directory domain for the account; and details about the account status, such as the date and time of the account's last logon, and whether the account is configured to expire, locked out, or disabled.	User Account Report
UsersReport.ps1	Lists information from the UNIX profile for each user in each zone. The report includes the user's Active Directory user name, UNIX user name, UID, shell, home directory, and primary group.	Users Report
WindowsUserEffectiveRightsReport.ps1	Lists the effective rights for each Windows user on each computer. The report shows the name of the right, its type, and where it is defined. This report applies to hierarchical zones only.	Hierarchical Zone – Windows User Effective Rights
ZoneDelegationReport.ps1	Lists the administrative tasks for each zone and the users or groups (trustees) that have been delegated to perform each task. When you grant administrative rights to designated users and groups, you make them "trustees" with permission to perform specific operations. This report indicates which users or groups have permission to perform specific tasks, such as add groups, join computers to a zone, or change zone properties.	Zone Delegation Report

This script	Reports this information	Is equivalent to this Access Manager Report Center report
ZoneRolePrivilegesReport.ps1	Lists the roles that are defined for each hierarchical zone and the rights granted by each of these roles, including where each right is defined.	Hierarchical Zone – Zone Role Privileges Report
ZonesReport.ps1	Lists the zone UNIX properties for each zone. This report includes the zone name, list of available shells, the default shell, the default home directory path, the default primary group, the next available UID, reserved UIDs, the next available GID, and reserved GIDs.	Zones Report

Running report scripts

When you perform the steps described in this section, the report content displays as text in the PowerShell console window. To generate formatted reports, see "Formatting reports."

To run a report script:

1. Open the Centrify Access Module for PowerShell Reports.
2. Verify you have permission to execute scripts.

   ```
   Get-ExecutionPolicy
   ```

   In most cases, the permission to execute scripts is restricted. You can use the `Set-ExecutionPolicy` to allow execution. For example:

   ```
   Set-ExecutionPolicy Unrestricted
   ```

   For more information about execution policies and the options available, use the `get-help` function.
3. Verify that you are in the directory where the report scripts are located. For example:

   ```
   C:\Program
   Files\Centrify\PowerShell\Centrify.DirectControl.PowerShell\R
   eports
   ```

4. Execute the report script. For example:

```
.\ZonesReport.ps1
```

Formatting reports

You can use the following cmdlets to format report output so that the output can be displayed or processed by third-party tools:

- `Export-Csv`
- `Out-GridView`
- `Format-Table`
- `ConvertTo-Html`

The following sections describe these cmdlets in detail.

Export-Csv

Use this cmdlet to format report output as a CSV file. For example, execute the following command to format the output from the `UsersReport.ps1` script as a CSV file:

```
PS> ./UsersReport.ps1 | Export-Csv C:\Report\UsersReport.csv -
NoTypeInformation
```

In this example, the output file `C:\Report\UsersReport.csv` is created, and no type information for the input object is provided. After the CSV file is created, it can be opened with third-party applications such as Microsoft Excel.

Out-GridView

Use this cmdlet to format report output as an interactive table in a grid view window. For example, execute the following command to format the output from the `UsersReport.ps1` script:

```
PS> ./UsersReport.ps1 | Out-GridView
```

Format-Table

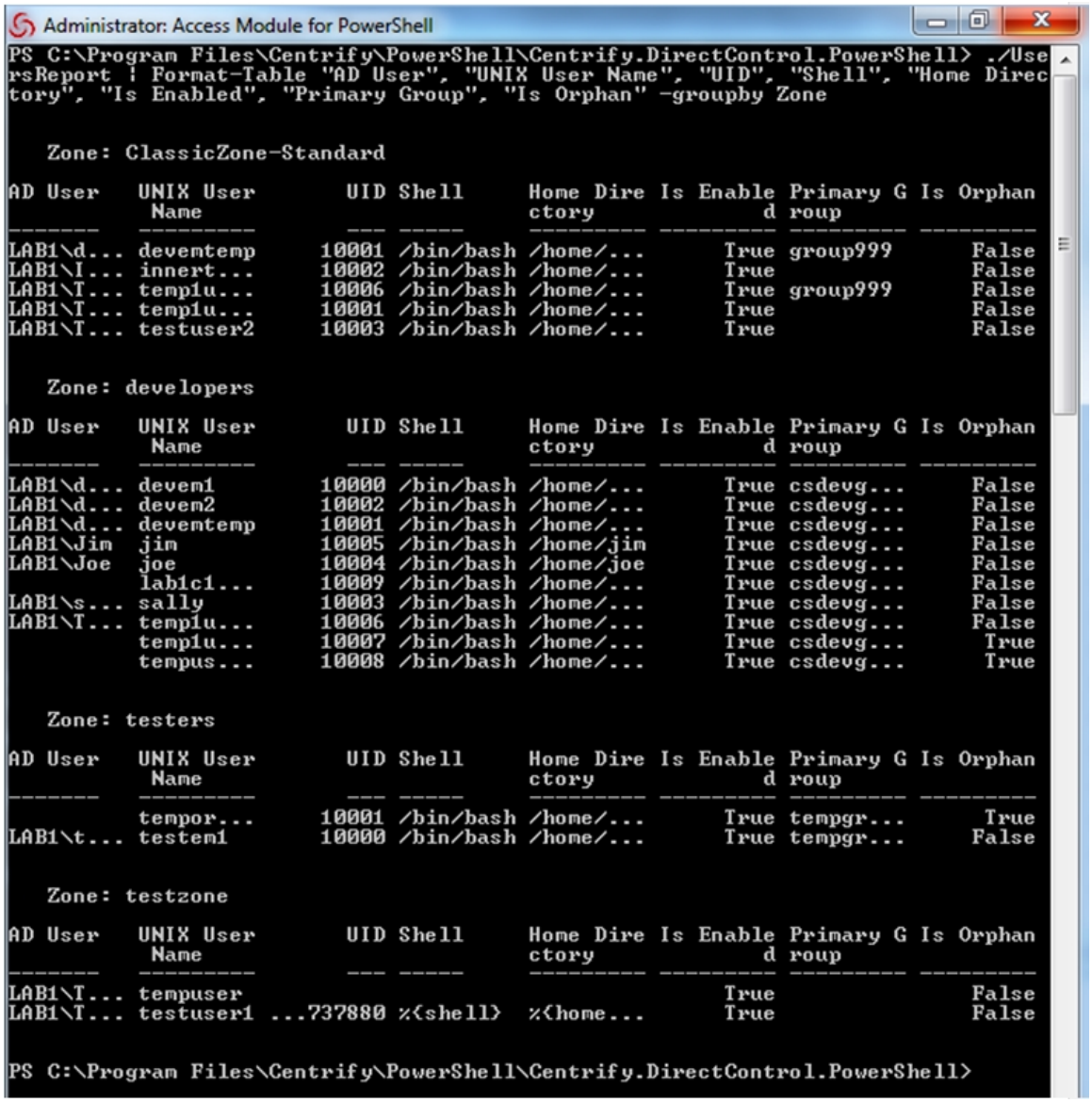
Use this cmdlet to format report output as a table that is displayed in the PowerShell console window, with the selected properties of the object in each column. The object type determines the default layout and properties that are displayed in each column, but you can use the Property parameter to select the properties that you want to display. You can specify any of the following parameters on the command line:

- Zone
- AD User
- UNIX User Name
- UID
- Shell
- Home Directory
- Is Enabled
- Primary Group
- Is Orphan

For example, the following command displays the output of `UsersReport.ps1` in a table. The `-GroupBy` option shown here specifies that separate tables are displayed for each zone. Each zone table contains columns for AD User, UNIX User Name, UID, Shell, Home Directory, Is Enabled, Primary Group, and Is Orphan.

```
. PS> ./UsersReport.ps1 | Format-Table "AD User", "UNIX User
Name", "UID", "Shell", "Home Directory", "Is Enabled", "Primary
Group", "Is Orphan" -GroupBy Zone
```

Depending on your site's zone configuration, this command would result in output similar to the following:



Note: If the results are too wide to display in PowerShell console default window size, you can change the PowerShell screen size, and enable some arguments (such as `wrap`, `autosize`, and so on) provided by this cmdlet.

ConvertTo-Html

Use this cmdlet to format report output as an HTML file. This cmdlet returns the result to the PowerShell console window. You can then redirect the result to an HTML file by using the cmdlet `Out-File`, so that you can read the output using a web browser. The HTML file created by this cmdlet uses the style sheet defined in the `report.css` file that is included with authentication and privilege elevation PowerShell.

For example, the following command converts the results of the `UsersReport.ps1` script into HTML using the style defined in `report.css`, and writes the resulting HTML to the output file `C:\Report\UsersReport.html`.

```
PS> .\UsersReport.ps1 | ConvertTo-Html -CssUri  report.css  |
Out-File C:\Report\UsersReport.html
```

Generating a PDF report

This section describes how to use the PDFCreator third-party tool to generate PDF output from a report script. The general steps in this procedure are as follows:

1. Install the PDFCreator third-party tool.
2. Generate HTML output from a report script using the `ConvertTo-Html` cmdlet.
3. Configure the PDFCreator printer that will convert the HTML output file into a PDF file.
4. Direct the HTML output file to the PDFCreator printer to generate the PDF file.

Procedure details

The following steps describe how to generate PDF output from the `ZonesReport.ps1` script.

- You must have administrator privileges to perform these steps.
- Unless otherwise noted, you perform the steps described here in the PowerShell console window.
- In this example, the PDF printer that converts HTML to PDF is named “PDFCreator”. If the printer has a different name in your environment, use your printer’s name.

1. Install PDFCreator from `http://www.pdfforge.org`.
2. Generate HTML output from the `ZonesReport.ps1` script by executing the following command in the PowerShell console:

   ```
   .\ZonesReport.ps1 | ConvertTo-Html -Head "<Style>$(Get-
   Content .\Report.css)</Style>" | Out-File
   c:\Reports\ZonesReport.html
   ```

When you execute this command, the file `c:\Reports\ZonesReport.html` is created using the styles in `Report.css`.

3. Specify PDFCreator as the default printer:

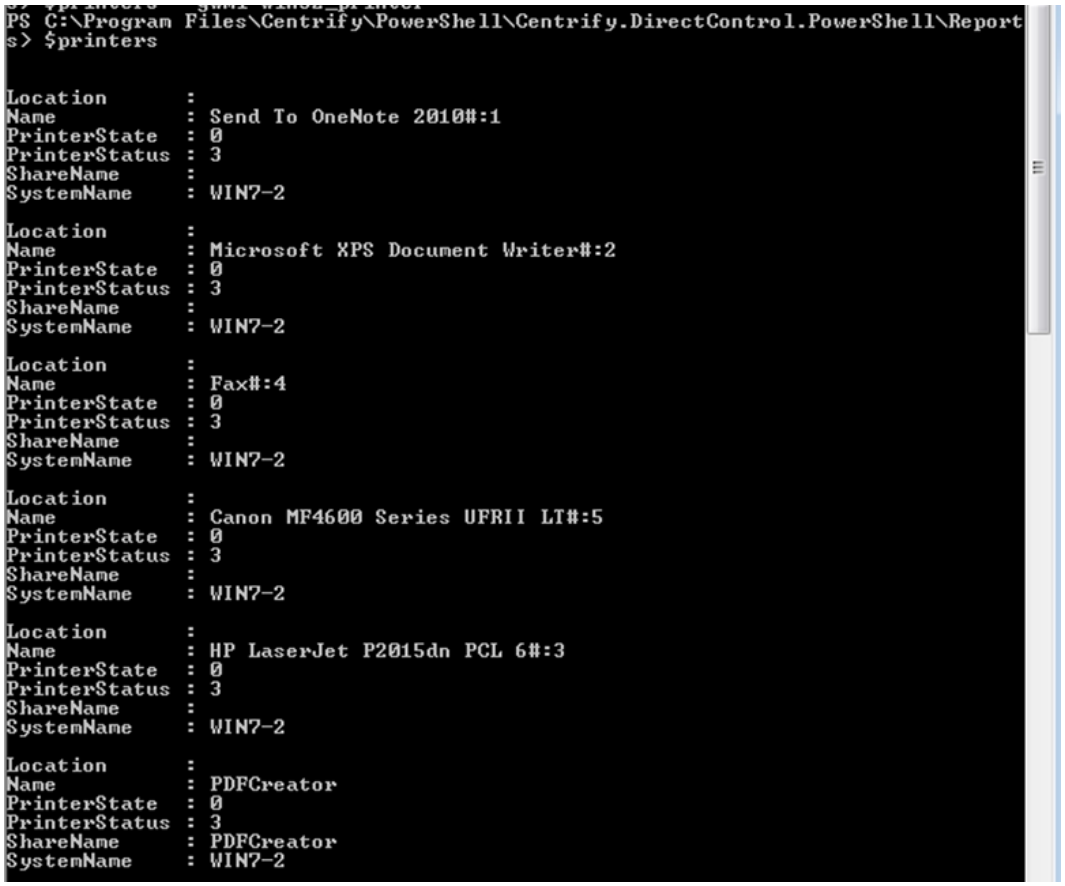
   a. Execute the following command to get all installed printers:

      ```
      $printers = gwmi win32_printer
      ```

   b. Run the following variable to list the printers:

      ```
      $printers
      ```

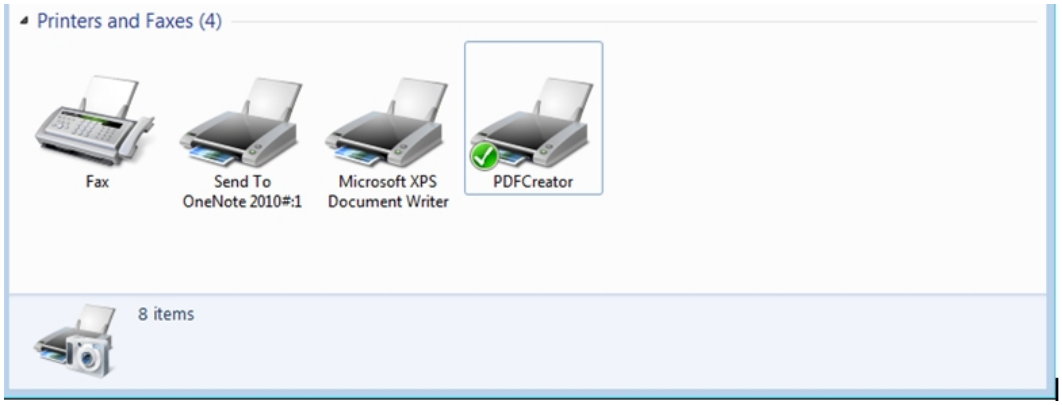
   c. In the list of printers, note the position of the PDFCreator printer in the list. For example, in the following list of printers, PDFCreator is the sixth printer listed:



   d. Make PDFCreator the default printer. In this example, because PDFCreator is the sixth printer on the list, you would execute the following command:

      ```
      $printers[5].SetDefaultPrinter()
      ```

e. Verify that PDFCreator is the default printer by opening the Devices and Printers control panel. If PDFCreator is not the default printer, you can make it the default printer here.



4. Configure auto-save printer settings as follows:
   - Change the auto-save directory to `C:\Reports`
   - Change the auto-save file name to `ZonesReport`
   - Enable the auto-save feature so that there will be no dialog prompts to ask for which file name to save

   Perform the following steps to configure the registry to implement these changes. These steps assume that the default registry path is `HKCU:\Software\PDFCreator\Program`. If your registry path is different, change these commands as appropriate for your environment.

   a. Execute the following command to change the auto-save directory to `C:\Reports`:

   ```
   Set-ItemProperty -Path
   "HKCU:\Software\PDFCreator\Program"
   -Name "AutoSaveDirectory" -Value "C:\Reports"
   ```

   b. Execute the following command to change the auto-save file name to `ZonesReport`:

   ```
   Set-ItemProperty -Path
   "HKCU:\Software\PDFCreator\Program"
   -Name "AutoSaveFileName" -Value "ZonesReport"
   ```

   c. Execute the following command to enable the auto-save feature:

   ```
   Set-ItemProperty -Path
   "HKCU:\Software\PDFCreator\Program" -Name "UseAutoSave"
   -Value "1"
   ```

5. Use Internet Explorer to print the HTML file that you created in Step 2 on the default (PDFCreator) printer. This step results in the creation of the

PDF file.

The recommended way to perform this step is to create and run the following script in the PowerShell console window. The script performs the following tasks:

- Creates an IE object and stores it into the variable `$ie`.
- Sets IE output not to display on the screen. This part of the script is optional; if you want IE output to display, you can omit this section of the script.
- Instructs the `$ie` object to read the HTML content from the location `C:\Reports\ZonesReport.html` (the HTML file that you created in Step 2).
- Prints the content of `$ie` using default printer (PDFCreator), resulting in the generation of the PDF file.

The recommended script is as follows:

```
$ie = New-Object -com "InternetExplorer.Application"
$ie.Visible = $false
$ie.Navigate("C:\Reports\ZonesReport.html")
while ( $ie.busy ) { Start-Sleep -second 1 }
$ie.ExecWB(6,2)
while ( $ie.busy ) { Start-Sleep -second 1 }
$ie.quit()
```

Note: This script is specific to the example used in this procedure. If you changed any of the steps in this procedure because of differences in your environment, you might have to make corresponding changes in the script shown here.