Centrify Server Suite 2014

Access Control and Privilege Management Scripting Guide

June 2014

Centrify Corporation

Legal notice

Contents

About this guide

The *Access Control and Privilege Management Scripting Guide* describes the Centrify DirectManage 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

The *Access Control and Privilege Management Scripting 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 Centrify DirectManage Access 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 Centrify DirectManage Access Manager console.

The guide provides the following information:

- Chapter 1, "Developing scripts for administrative tasks," provides an introduction to access control and privilege management using Windows PowerShell.

- Chapter 2, "Installing the access module for PowerShell," describes how to download and install the module as a separate package.
- Chapter 3, "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.
- Chapter 4, "Objects and properties," lists the objects defined by the Centrify DirectManage PowerShell module, and the properties of each object.
- Chapter 5, "Adding users in a one-way trust environment," explains how to add a user in a one-way trust environment by using the Centrify DirectManage PowerShell module.

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. 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 Centrify DirectManage Access 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 DirectManage, consult the Centrify Web site or Centrify Support to find out if another version of this guide is available. Because the Centrify DirectManage Access cmdlets are subject to change, enhancement, or replacement, the information in this guide can also become incomplete, obsolete, or unsupported in future versions.

Conventions used in this guide

The following conventions are used in this guide:

- `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`*, the fixed-width font is used to indicate variables.
- **Bold** text is used to indicate commands, buttons, or user interface text.
- *Italics* are used for book titles and to emphasize specific words or terms

Finding information about Centrify products

Centrify Server Suite includes extensive documentation targeted for specific audiences, functional roles, or topics of interest. However, most of the information in the documentation set is intended for administrators, application developers, or security architects after you have purchased the software or licensed specific features. 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.

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 customer support

If you have a Centrify account, click Support on the Centrify website to log on and access the Centrify Customer Support Portal. From the support portal, you can search knowledge base articles, open and view support cases, connect with other Centrify users on customer forums, and access additional resources—such as online training, how-to videos, and diagnostic tools.

Developing scripts for administrative tasks

The Centrify DirectManage Access Module 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.

On Windows computers, you can use the Centrify DirectManage Access 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 *Centrify Server Suite ADEdit Command Reference and Scripting Guide*.

Writing programs in other languages

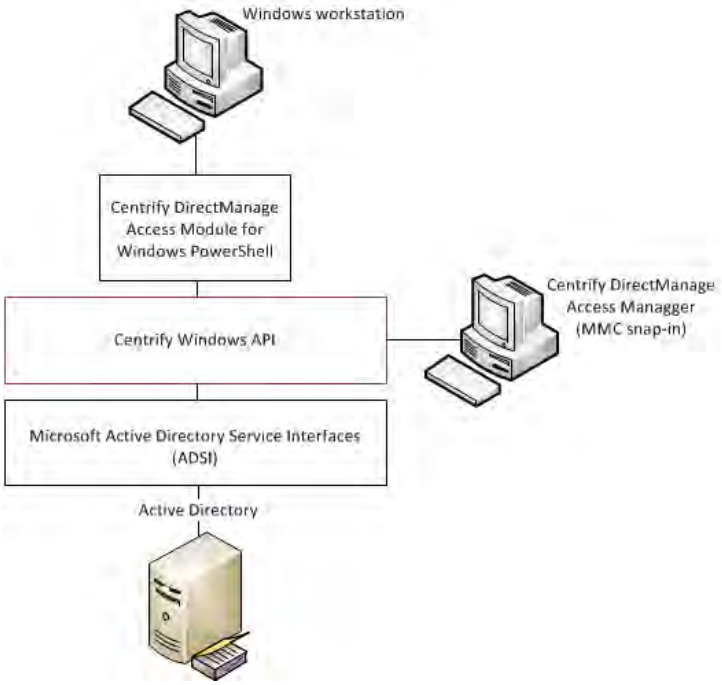
If you want to develop programs or scripts that run on Windows but outside of the Windows PowerShell environment, you can use any language that supports the Component Object Model (COM) interface. The Centrify COM-based interface is available as part of the Centrify Windows Software Development Kit (SDK). 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 *Centrify Windows API Programmer's Guide*.

Accessing information stored in Active Directory

The Centrify DirectManage Access Module for Windows 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 Centrify DirectManage 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 DirectManage Access Module for Windows 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 DirectManage Access Module for Windows 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 Centrify DirectManage Access module for PowerShell from the Centrify Server Suite 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.

The following topics are covered:

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 Centrify DirectManage Access Manager console. You can also download these libraries separately, along with sample scripts and documentation, onto computers where Centrify DirectManage Access Manager is not installed.

If you are downloading the Access Module for PowerShell as a separate software package, you can choose the operating environment on which you plan to develop. For example, if you only want to develop scripts that run on 32-bit Windows computers, you can download a single compressed folder that contains the package for Windows 32-bit computers. If you are developing scripts to run on 64-bit Windows computers, you should download the compressed folder that contains the package for Windows 64-bit computers.

You can download the Access Module for PowerShell as a separate package from the Centrify Customer DownLoad 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 32-bit or 64-bit 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 Centrify DirectManage Access 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 Access module

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 Centrify DirectManage Access cmdlets are available in a separate Centrify DirectManage Access Module for Windows PowerShell console.

   If you want the Centrify DirectManage Access 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 Centrify DirectManage Access 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 Centrify DirectManage Access module into your default Windows PowerShell console.

To import the Centrify DirectManage Access 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 Centrify DirectManage Access 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 Centrify DirectManage Access 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 Centrify DirectManage Access 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.

The following topics are covered:

Using cmdlets to manage access

The Centrify DirectManage 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-Cdm`*Xxx* cmdlets create new Centrify objects, such as a new zone or a new role definition.
- `Add-Cdm`*Xxx* cmdlets add a right to a specified role.
- `Get-Cdm`*Xxx* cmdlets get the properties of a specified object.
- `Set-Cdm`*Xxx* cmdlets set or change the properties of a specified object.
- `Remove-Cdm`*Xxx* 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 DirectManage 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
TargetTypeUser
--------------
finance.acmeForestadministrator@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-Cdm`*`Xxx`* cmdlet.

For most `Set-Cdm`*`Xxx`* or `Remove-Cdm`*`Xxx`* cmdlets, you must call the corresponding `Get-Cdm`*`Xxx`* or `Add-Cdm`*`Xxx`* 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 Centrify DirectManage Access 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 DirectManage 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.";
```

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.
`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.
`MatchCriteria`	`MatchCriteria array`	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.

Property	Type	Description
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.
IsPreserveGroup	Boolean	True to retain the user's group membership while executing a command.
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.
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.
`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
`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.
`Name`	`string`	Name of the desktop right.
`PreferredServer`	`string`	Preferred server to use for committing changes to Active Directory.
`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.

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.

CdmManagedComputer

Represents a computer managed by Centrify DirectManage Access. 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.

Property	Type	Description
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.
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.
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.

Property	Type	Description
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.
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.
Name	string	Name of the network right.

Property	Type	Description
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 Centrify DirectManage Access 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.
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.

Property	Type	Description
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 Centrify DirectManage Access 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.
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.

Property	Type	Description
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.
IsSfu	Boolean	True if the user profile is an SFU profile.

Property	Type	Description
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.
Description	string	Description of the zone.
DistinguishedName	string	Distinguished name of the zone.
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.
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.
Schema	string	Schema of the zone.
SfuDomain	string	SFU domain of the zone (SFU zone only).

Property	Type	Description
`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.